



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**REKONSTRUKCE POŠKOZENÉ ČÁSTI OTISKU PRSTŮ  
S VYUŽITÍM NEURONOVÝCH SÍTÍ**

RECONSTRUCTION OF DAMAGED PARTS OF FINGERPRINT IMAGE USING NEURAL NETS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. MICHAEL HALINÁR**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ONDŘEJ KANICH, Ph.D.**

BRNO 2020

## Zadání diplomové práce



Student: **Halinár Michael, Bc.**  
Program: Informační technologie Obor: Inteligentní systémy  
Název: **Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí**  
**Reconstruction of Damaged Parts of Fingerprint Image Using Neural Nets**  
Kategorie: Zpracování obrazu

### Zadání:

1. Prostudujte literaturu týkající biometrického rozpoznávání osob podle otisků prstů. Soustřed'te se na metody zvyšující kvalitu otisku. Seznamte se s metodami pro rekonstrukci, případně generování syntetických, otisků prstů a detekci poškození. Prostudujte možnosti neuronových sítí v oblasti generování syntetických otisků a rekonstrukci otisků prstů.
2. Navrhněte metodu pro rekonstrukci otisku prstu v poškozených oblastech s využitím neuronových sítí.
3. Implementujte navržené algoritmy z předchozího bodu.
4. Otestujte algoritmy na vhodné databázi (např. syntetických otisků). Analyzujte přesnost provedené rekonstrukce.
5. Dosažené výsledky shrňte a diskutujte. Uveďte možná rozšíření Vašeho řešení.

### Literatura:

- Minaee, S., Abdolrashidi, A.: *Finger-GAN: Generating Realistic Fingerprint Images Using Connectivity Imposed GAN*, Preprint, 2018.
- Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer, 2009, pages 512. ISBN 978-1-8488-2254-2.
- Dejmal, D.: *Analysis of Fingerprint Quality Assessment Systems*, Brno, 2019. Bachelor's thesis. FIT BUT, in Brno.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Kanich Ondřej, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 3. června 2020

Datum schválení: 31. října 2019

## Abstrakt

Táto práca sa zaoberá problematikou rekonštrukcie odtlačkov prstov pričom kladie dôraz na použitie neurónových sietí. Najprv je analyzovaná štruktúra odtlačku prstu, nasleduje popis možností, ktorými sa dá zvyšovať kvalita odtlačku. Jednoduchý úvod do neurónových sietí čitateľovi poskytne obraz o použitých technológiách. Po výbere architektúry neurónovej siete nasleduje proces jej učenia. Bolo vytvorené aj jednoduché grafické rozhranie pre prácu s odtlačkami. Aplikácia dokáže rekonštruovať syntetické odtlačky prstov poškodené bradavicami rôznych veľkostí. Pomocou ďalšej neurónovej siete je možné detegovať lokáciu bradavice. Testy úspešnosti rekonštrukcie preukázali zlepšenie kvality o 43,5 % v prípade odtlačku poškodeného desiatimi bradavicami. Porovnávacie skóre odtlačkov pred a po rekonštrukcii sa zvýšilo na tejto poškodenej sade o 6,5 %.

## Abstract

This thesis deals with the issue of reconstruction of damaged fingerprints using artificial neural networks. At first, the fingerprint structure is analyzed, after that, the methods that can be used to improve fingerprint quality are described. An introduction to neural networks is given for understanding the basics of artificial neural networks. After choosing the right architecture for the neural networks, the process of its learning is described. A simple graphic user interface was created for this application, which is able to reconstruct synthetic fingerprints damaged by various warts. Another neural net can detect the location of wart. Tests have proven an increase in the quality of fingerprint by 43,5 % in the dataset with ten inserted warts on each fingerprint. The matching score was increased by 6,5 % on this particular dataset.

## Kľúčové slová

konvulučná neurónová sieť, odtlačok prsta, rekonštrukcia odtlačku prsta, syntetický odtlačok prsta, bradavica, spracovanie obrazu

## Keywords

convolutional neural network, fingerprint, fingerprint reconstruction, synthetic fingerprint, wart, image processing

## Citácia

HALINÁR, Michael. *Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Kanich, Ph.D.

# Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí

## Prehlásenie

Prehlasujem túto diplomovú prácu som vypracoval samostatne pod vedením pána doktora Ondřeje Kanicha. Uviedol som všetky literárne pramene z ktorých som čerpal.

.....

Michael Halinár

3. júna 2020

## Podakovanie

Chcel by som poďakovať môjmu vedúcemu práce Ondřejovi Kanichovi za poskytnuté materiály a pomoc pri vytváraní práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Rozpoznávanie osôb podľa odtlačkov prstu</b>	<b>4</b>
2.1	Štruktúra odtlačku . . . . .	4
2.2	Vlastnosti odtlačkov . . . . .	4
2.3	Frekvencia línií . . . . .	5
2.4	Orientácia línií . . . . .	6
2.5	Snímanie odtlačkov prstov . . . . .	6
<b>3</b>	<b>Zvyšovanie kvality odtlačku prstu</b>	<b>11</b>
3.1	Normalizácia . . . . .	12
3.2	Kontextové filtre . . . . .	12
3.3	Ochorenie pokožky . . . . .	13
3.4	Kvalita a verifikácia . . . . .	15
<b>4</b>	<b>Neurónové siete</b>	<b>17</b>
4.1	Strojové učenie . . . . .	17
4.2	Biologický neurón . . . . .	18
4.3	Umelý neurón . . . . .	18
4.4	Umelá neurónová sieť . . . . .	18
4.5	Konvolučná neurónová sieť . . . . .	19
4.6	Postupy učenia siete . . . . .	21
<b>5</b>	<b>Generovanie syntetických odtlačkov prstov a simulácia poškodenia</b>	<b>24</b>
5.1	SFinGe . . . . .	24
5.2	Anguli . . . . .	25
5.3	SYnFI . . . . .	26
5.4	SyFDaS . . . . .	27
<b>6</b>	<b>Návrh</b>	<b>28</b>
6.1	Získanie a úprava odtlačkov . . . . .	28
6.2	Výber architektúry . . . . .	29
6.3	Výber knižnice . . . . .	30
6.4	Úprava siete, učenie a rekonštrukcia . . . . .	32
<b>7</b>	<b>Implementácia</b>	<b>33</b>
7.1	Vytváranie databáze odtlačkov . . . . .	33
7.2	Učenie siete . . . . .	36

7.3 Grafické užívateľské rozhranie . . . . .	40
<b>8 Vyhodnotenie úspešnosti rekonštrukcie</b>	<b>42</b>
8.1 NFIQ2 . . . . .	42
8.2 FiQiVi . . . . .	44
8.3 VeriFinger . . . . .	44
<b>9 Záver</b>	<b>47</b>
<b>Literatúra</b>	<b>48</b>
<b>A Obsah priloženého pamäťového média</b>	<b>51</b>
<b>B Grafy úspešnosti rekonštrukcie</b>	<b>52</b>

# Kapitola 1

## Úvod

Rozpoznávanie podľa odtlačkov prstu bolo zavedené už v 19. storočí [4]. S nástupom digitálnej doby sa rapídne zvýšil záujem využívania odtlačkov ako spôsob identifikácie a verifikácie osôb. Jednoduché a pohodlné používanie tejto technológie na identifikáciu osôb je veľkou výhodou. V dnešnej dobe sa stretávame so snímačmi odtlačkov prstov takmer každodenne. Snímače nachádzame v mobiloch, notebookoch či iných zariadeniach.

Kvalita odtlačkov nie je vždy dostačujúca. Existuje veľa príčin, kvôli ktorým môžu byť odtlačky osoby poškodené. Od rezných rán po rôzne ochorenia kože, medzi ktoré patria napríklad bradavice a ekzémy. Ďalšou príčinou nekvalitného odtlačku môže byť nesprávne zaobchádzanie so snímačom odtlačkov prstov. Napríklad špinavý senzor či neprimeraný tlak na snímač vytvoria v snímke odtlačku šum. Preto je žiaduce vytvárať softvér, ktorý dokáže poškodené odtlačky rekonštruovať a zvyšovať ich kvalitu.

Cieľom tejto práce je naštudovanie problematiky odtlačkov prstov a návrh aplikácie, ktorá bude rekonštruovať odtlačky s využitím neurónových sietí. Do odtlačkov bude vkladané poškodenie vo forme niekoľkých bradavíc. Neurónová sieť bude optimalizovaná pre rekonštruovanie tohto poškodenia do pôvodného stavu. Vyhodnotenie výstupov bude prebiehať meraním kvality odtlačku a porovnávacieho skóre. Po rekonštrukcii odtlačku by sa mali tieto hodnoty zvýšiť, ideálne až na úroveň originálneho nepoškodeného odtlačku. Zlepšenie kvality odtlačku však nemusí vždy znamenať zlepšenie porovnávacieho skóre.

V druhej kapitole je analyzovaná štruktúra odtlačku prstu. Ďalej je pozornosť zameraná na vlastnosti a typické vzory nachádzajúce sa v odtlačkoch, vďaka ktorým môžeme odtlačky klasifikovať a porovnávať. V ďalšej kapitole je popísaný proces získavania odtlačku pomocou snímača odtlačkov prstov. Tretia kapitola sa zameriava na zvyšovanie kvality odtlačkov prstov pomocou normalizácie a kontextových filtrov. Keďže v tejto práci sú využívané syntetické odtlačky, ďalšia kapitola popisuje postup a dostupné nástroje pre vytváranie takýchto odtlačkov. Nasleduje kapitola venovaná neurónovým sieťam. Obsahuje jednoduchý úvod do problematiky neurónových sietí začínajúc biologickým neurónom až po konkrétne architektúry sietí. Po kapitole popisujúcej návrh a nasleduje implementácia aplikácie s grafickým rozhraním. Táto aplikácia je schopná rekonštruovať bradavicami poškodené syntetické odtlačky prstov. V kapitole obsahujúcej testy sú analyzované výsledky práce. V závere sú zhrnuté dosiahnuté výsledky ale aj možnosti ďalšieho rozšírenia aplikácie.

## Kapitola 2

# Rozpoznávanie osôb podľa odtlačkov prstu

V tejto kapitole bude popísaná základná štruktúra odtlačku prstu, ďalej budú vymenované vlastnosti odtlačkov prstov, ktoré sú kľúčové pre analýzu a použitie pri identifikácii osôb. Taktiež bude opísaný senzor, vďaka ktorému je možné získavať odtlačky prstov.

### 2.1 Štruktúra odtlačku

Odtlačok prsta vzniká odtlačením papilárnych línií na určitý povrch. Forma papilárnych línií je určená hlavne génmi jedinca [1]. Ak nepríde k vážnemu mechanickému alebo inému poškodeniu, línie zostávajú rovnaké počas celého života, pokiaľ zanedbáme faktor starnutia kože, ktorý je prirodzený. Papilárne línie sú formované v nižšej vrstve kože čo zabezpečuje ich obnovu v prípade ľahkého poškodenia [6]. Línie vytvárajú vzory, vďaka ktorým môžeme odtlačky jednoznačne identifikovať.

### 2.2 Vlastnosti odtlačkov

Ak chceme odtlačok analyzovať a použiť ho pre identifikáciu, musí obsahovať významné rysy a musí byť vhodný pre ďalšiu analýzu. Významné rysy znamenajú prítomnosť jednoznačne rozlíšiteľných vlastností. Vhodnosť odtlačku znamená že tieto vlastnosti môžu byť jednoducho extrahované z reprezentácie odtlačku, uložené a použiteľné pre porovnanie odtlačkov. Obrázková reprezentácia odtlačku nie je vždy vhodná kvôli variáciám v jase, kvalite obrázku, chybám a ďalším rôznym globálnym skresleniam. Ďalším problémom, v dnešnej dobe nie až tak významným, je veľkosť potrebnej pamäte na uchovanie obrazového odtlačku. V odtlačkoch je preto nutné hľadať vzory. V odtlačkoch môžeme pozorovať 3 úrovne vzorov. Na najväčšej globálnej úrovni môžeme pozorovať vzory podobné ako v obrázku 2.1. Tieto vzory sa nazývajú singulárne body, patrí medzi ne delta a jadro, na obrázku sú označené trojuholníkom a štvorcom. Tieto vlastnosti sa používajú na klasifikáciu odtlačkov. Nie sú vhodné pre porovnanie odtlačkov pretože sú málo unikátne. Medzi ďalšie globálne vlastnosti patria tvar prstu, orientácia, veľkosť. [6]

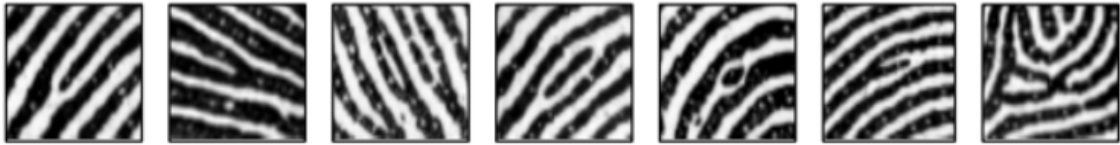
V druhej úrovni môžeme pozorovať niekoľko rôznych charakteristík papilárnych línií. Nie sú rovnomerne distribuované a závisia na kvalite odtlačku. Vo všeobecnosti sú veľmi vhodnou a stabilnou voľbou pre porovnanie odtlačkov. Ich automatická detekcia môže byť zložitá pri nízko kvalitných obrázkoch. Nazývame ich markanty. Na jednoznačné urče-





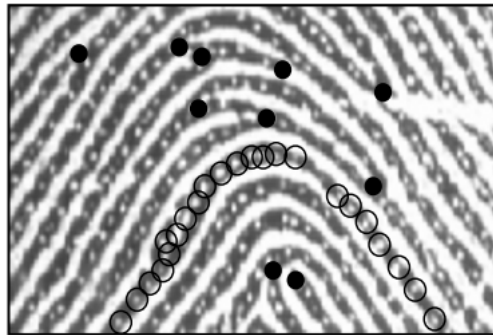
Obrázok 2.1: Vzory najvyššej úrovne, prevzaté z [23].

nie vlastníka odtlačku s vysokou pravdepodobnosťou stačí iba 5 až 7 markantov, napríklad pomocou nástroja VeriFinger [25]. Markant je určený pozíciou čiže súradnicami  $x$  a  $y$ , ďalej uhlom  $\theta$  smeru markantu a jeho typom. Medzi hlavné typy markantov patria: ukončenie, dvojité vidlička, jednoduchá slučka, interval, bod, hák, kríženie. Jednotlivé typy markantov môžeme vidieť na obrázku 2.2.



Obrázok 2.2: Hlavné typy markantov, prevzaté z [23].

V tretej úrovni môžeme pozorovať veľmi jemné vzory, medzi ktoré patria napríklad šírka línií, tvar, zakrivenie, obrys hrán. Najdôležitejšími sú však potné póry znázornené na obrázku 2.3, pre ich vysoké rozlišovacie vlastnosti. Avšak takéto jemné vzory môžeme pozorovať iba vo veľmi kvalitných obrázkoch. Preto nie sú vhodné pre bežné použitie. Svoje využitie si ale nájdu vo forenznej analýze.

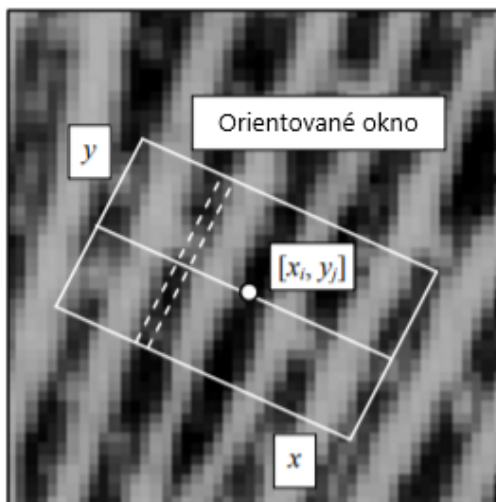


Obrázok 2.3: Potné póry a markanty v papilárnych líniách, prevzaté z [23].

## 2.3 Frekvencia línií

Lokálna frekvencia línií  $f_{xy}$  je počet línií v jednotke dĺžky, vo vnútri orientovaného okna so stredom v bode  $[x, y]$ , kolmo na smer línií. Existuje viacero možností ako frekvenciu počítať. Jednoduchým a rýchlym spôsobom je výpočet priemerného počtu pixelov medzi

dvoma súbežne idúcimi vrcholmi línií pozdĺž ich orientácie. Orientované okno je znázornené na obrázku 2.4. [23]



Obrázok 2.4: Výpočet frekvencie línií v orientovanom okne, prevzaté a upravené z [23].

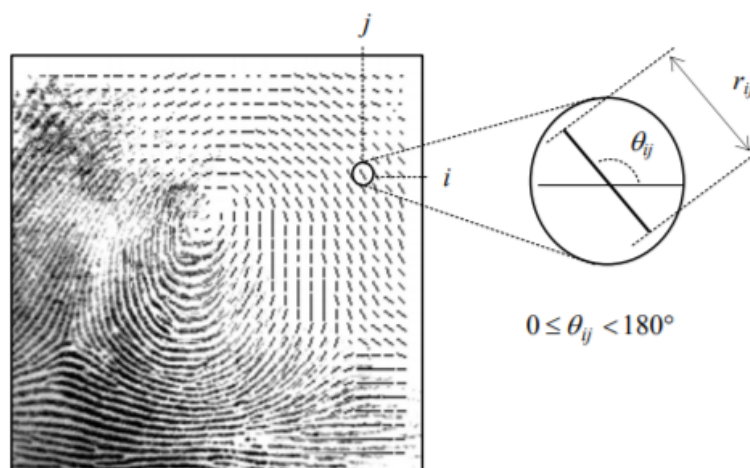
## 2.4 Orientácia línií

Lokálna orientácia línií je uhol  $\theta_{xy}$ , ktorý zvierajú papilárna línia s horizontálnou osou v určitej oblasti pixelov so stredom v bode  $[x, y]$ . Pretože papilárne línie nie sú orientované, uhol  $\theta_{xy}$  môže nadobúdať hodnotu rozpätia  $0^\circ$  až  $180^\circ$ . Orientáciu línie znázorňuje obrázok 2.5. Namiesto počítania orientácie v každom pixeli, väčšina algoritmov počíta orientáciu v určitých blokoch obrázka za účelom redukcie ceny výpočtu. Takýmto výpočtom nad celým obrázkom dostávame obrázok orientácií. Je to matica, ktorej elementy popisujú lokálnu orientáciu papilárnych línií. Každý jej prvok  $[i, j]$  popisuje priemernú orientáciu línií v okolí bodu  $[x_i, y_j]$ . Ďalšou hodnotou, ktorá sa nachádza v takejto matici je  $r_{ij}$ , tá určuje konzistenciu orientácie. Veľká hodnota tohoto parametru značí oblasť obrázka s vysokou kvalitou, naopak nízka hodnota určuje poškodené a zašumené oblasti obrázka [23].

## 2.5 Snímanie odtlačkov prstov

V minulosti sa využívala technika snímania prstov, ktorá spočíva v použití atramentu. Daná osoba si nanesie atrament na prst a následne ho otláči na papier. Tento papier je potom nasnímaný skenerom, čím vznikne digitálna verzia odtlačku prstu. Takéto snímanie sa označuje ako *offline* snímanie. Ďalší druh *offline* snímání sa využíva pri zbere odtlačkov v miestach činu forenznými expertmi. V dnešnej dobe sa prevažne používa snímanie odtlačkov pomocou elektronických senzorov, taktiež nazývaných ako čítačky odtlačkov prstov. Na takéto získavanie nie je potrebný atrament a stačí aby osoba priložila prst na čítačku. Snímače odtlačkov prstov môžu byť rozdelené do dvoch kategórií podľa počtu snímaných prstov:

- Snímače jedného prstu: malá plocha senzoru umožňuje snímanie iba jedného prstu v daný moment. Veľkými výhodami takýchto snímačov sú cenová dostupnosť, kom-



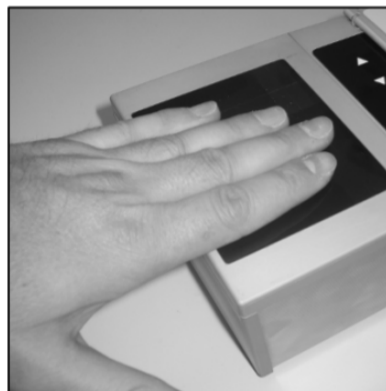
Obrázok 2.5: Orientácia papilárnych línií, prevzaté z [23].

paktné rozmery a jednoduchosť používania. Takéto snímače nájdu využitie v mnohých zariadeniach ako napríklad notebooky či iné mobilné zariadenia, snímač jedného prstu je znázornený na obrázku 2.6a.

- Snímače viacerých prstov: plocha senzoru je väčšia, umožňujúca snímanie viacerých prstov naraz. Takéto snímače dokážu získať všetkých 10 odtlačkov prstov v troch snímaniach, palce sa snímajú naraz, potom ostatné prsty jednej a druhej ruky. Nasleduje segmentácia jednotlivých odtlačkov, ktorá je zvyčajne robená automaticky, softvérom na to určeným. Výhoda spočíva v rýchlosti, ktorou dokáže snímač získať všetky odtlačky prstov, viď obrázok 2.6b.



(a)



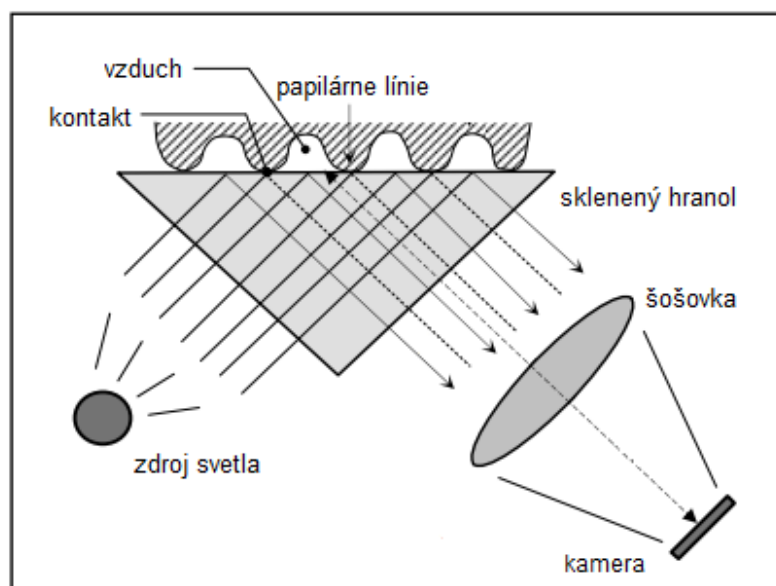
(b)

Obrázok 2.6: Snímač jedného prstu (a), snímač viacerých prstov (b), prevzaté z [23].

Niektoré snímače dokážu získavať takzvané rolované odtlačky. Daná osoba postupne roluje svoj prst po senzore tak aby zachytil čo najväčšiu nasnímanú plochu. Takto získaný odtlačok obsahuje väčšie množstvo informácií, oproti klasickému odtlačku. Niekedy je potrebné aby pri odbere odtlačku asistoval školený človek. [23]

### 2.5.1 Optický snímač

Najdôležitejšia časť čítačky odtlačkov prstov je senzor. V tejto časti snímača sa vytvára obraz odtlačku prstu. Jedným z najpoužívanejších a najstarších druhov senzoru je optický senzor, znázornený na obrázku 2.7. Pri dotyku prsta na sklenený povrch sa papilárne línie sklad dotýkajú ale medzi ryhami a sklom je vzduch. Pri nasvietení skleneného hranola, papilárne línie pohltia svetlo a v nasnímanom obrázku sa javia ako tmavé oblasti, na rozdiel od rýh, ktoré nie sú zaznamenané. [23] [7]



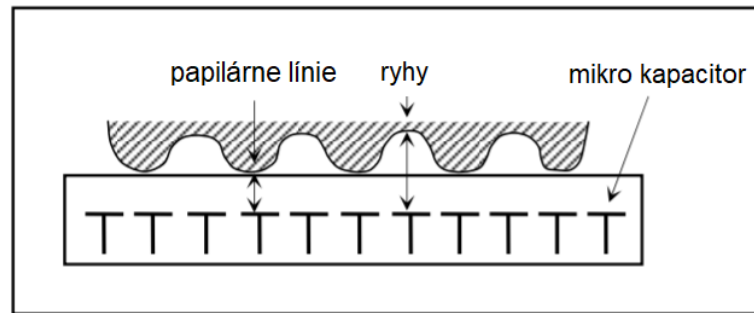
Obrázok 2.7: Optický snímač, prevzaté a upravené z [23].

### 2.5.2 Kapacitný snímač

Kapacitný snímač je taktiež jeden z najpoužívanejších. Tento snímač je tvorený maticou mikro-kapacitných plôch uložených na čipe. Po priložení prsta na snímač, vzniká elektrický náboj medzi povrchom prsta a jednou z mikro-kapacitných plôch. Sila náboja potom závisí od vzdialenosti prsta a plochy senzoru. Rozdielne vzdialenosti rýh a papilárnych línií teda znamenajú rozdielne veľkosti elektrického náboja. Na základe týchto rozdielov potom môžeme detegovať vzory na odtlačkoch prstoch. Jedným z výhod takéhoto snímača je možnosť úpravy jeho parametrov, v prípade že poskytnutý odtlačok nevykazuje najlepšie známky kvality spôsobené mierou vlhkosti prsta. Schéma kapacitného snímača je znázornená na obrázku 2.8. [23] [7]

### 2.5.3 Termický snímač

Termický snímač obsahuje pyro-elektrické senzory, ktoré vytvárajú elektrický prúd v závislosti tepla. Čím viac tepla je dodaného na senzor tým väčší elektrický prúd je generovaný. V oblasti papilárnej línie, ktorá je v kontakte so senzorom je teda generovaný väčší prúd ako v oblasti ryhy. Sledovaním väčšieho prúdu môžeme teda rozlišovať ryhy a línie. Tento snímač býva nahrievaný za účelom zväčšenia teplotných rozdielov v oblastiach línií. Výho-

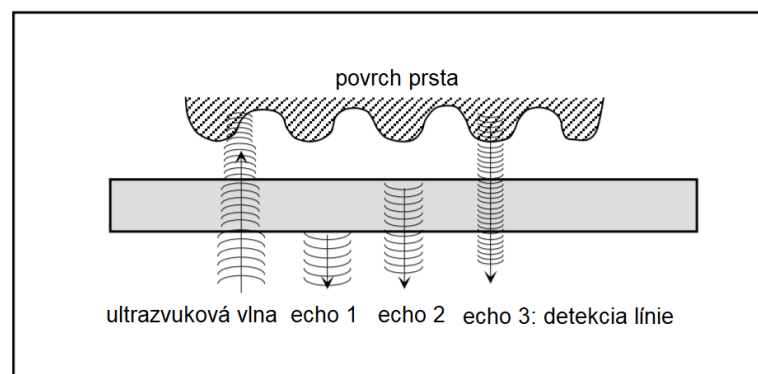


Obrázok 2.8: Kapacitný snímač, prevzaté a upravené z [23].

dou termického senzora je hrúbka ochranného obalu senzora, ktorý môže byť hrubší oproti ostatným sensorom, nakoľko teplo dobre prechádza aj cez takýto obal. [23] [7]

#### 2.5.4 Ultrazvukový snímač

Tento druh senzora využíva ultrazvukové vlny pre detekciu papilárnych línií. Sensor sa skladá z dvoch častí: vysielač a prijímač. Vysielač vytvára zvukové vlny a smeruje ich na povrch priloženého prsta. Odrazená zvuková vlna od povrchu je detegovaná prijímačom. Z časového oneskorenia signálu je potom možné vypočítať vzdialenosť povrchu, od ktorého sa vlna odrazila. Z rozdielov vzdialeností, línií a rýh je potom možné získať štruktúru povrchu prsta. Výhodou takéhoto snímača je že zvukové vlny dokážu prejsť cez tenký materiál, to znamená že v prípade nečistôt na snímači alebo pokožke je stále možné získať kvalitný odťahok avšak za cenu dlhšej doby snímania oproti ostatným druhom snímačov. Schému ultrazvukového snímača znázorňuje obrázok 2.9. [23] [7]

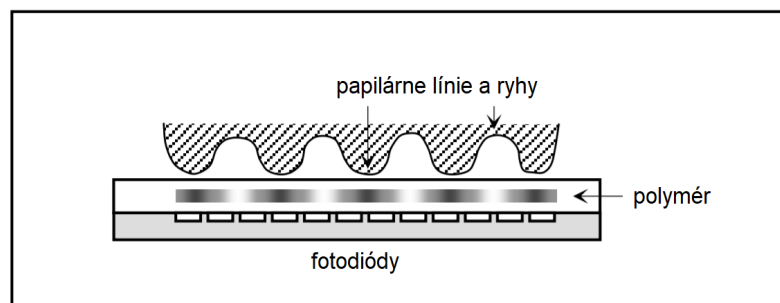


Obrázok 2.9: Ultrazvukový snímač, prevzaté a upravené z [23].

#### 2.5.5 Elektro-optický snímač

Elektro-optický snímač sa skladá z dvoch častí. V prevej časti sa nachádza polymér, ktorý po kontakte s pokožkou vyžaruje svetlo. Druhá časť obsahuje maticu fotodiód, ktoré prijímajú vytvorené svetlo. V závislosti od prijatého svetla na jednotlivých fotodiódach je potom

možné vytvoriť obraz štruktúry prsta. V niektorých verziách tejto technológie sa používa namiesto fotodiód jednoduchá kamera, ktorá zabezpečí získanie obrazu. Schéma elektro-optického snímaču je znázornená na obrázku 2.10. [23] [7]



Obrázok 2.10: Elektro-optický snímač, prevzaté a upravené z [23].

## Kapitola 3

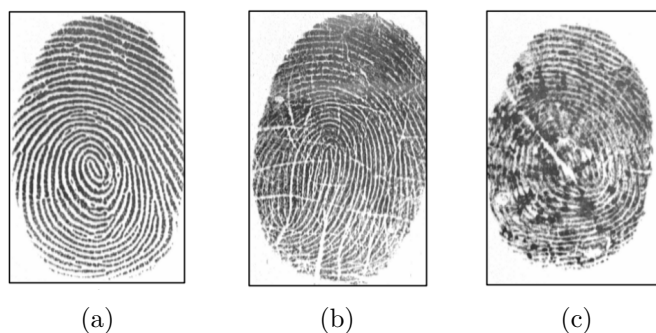
# Zvyšovanie kvality odtlačku prstu

Kvalita obrázku odtlačku prstu je kľúčová pri operáciách s daným odtlačkom. Ak by sme zobrali do úvahy ideálny odtlačok prsta, dostali by sme kvalitný obrázok vo vysokom rozlíšení, kde papilárne línie nie sú ničím prerušované a sú jednoznačne rozlíšiteľné. Z dôvodu rôznych ochorení kože, porezania, špinavého senzora, či príliš mokrého alebo suchého prstu je úloha dostať takýto kvalitný odtlačok pomerne náročná.

Rozlišujeme niekoľko typov degradácie v obrázkoch odtlačkov prstov:

- Línie nie sú spojité dôsledkom nekvalitného snímania,
- paralelné línie nie sú oddelené dôsledkom šumu,
- línie sú prerušené kvôli poraneniam, a chorobám prsta.

Tieto nedostatky nekvalitného obrázka spôsobujú rozpoznávanie markantov náramne zložité. Vznikajú tu problémy kedy môžu byť detegované chybné markanty, ostatné môžu byť zase prehliadnuté alebo vzniká chyba v orientácii či pozícii markantu. Rôzne druhy kvality odtlačkov sú znázornené na obrázku 3.1. Z dôvodu korektného rozpoznania markantov je nutné obrázok nejakým spôsobom upraviť tak aby sme získali čo najlepšiu reprezentáciu papilárnych línií, tak ako zodpovedajú realite.



Obrázok 3.1: Príklady rôznej kvality odtlačkov prstov: kvalitný (a) , stredne kvalitný odtlačok (b), nekvalitný (c), prevzaté z [23].

V súvislosti s kvalitou obrázku môžeme pozorovať na danom odtlačku tri druhy oblastí a to: dobre špecifikovaná, obnoviteľná a nepoužiteľná oblasť. U dobre špecifikovanej oblasti sú línie jednoznačné a nie je potrebná ich úprava. Obnoviteľné oblasti majú nižšiu kvalitu avšak použitím vhodnej metódy môžeme oblasť upraviť tak aby bola použiteľná. V závislosti

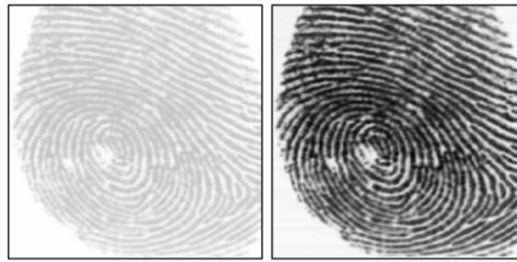
od aplikácie je možné oblasť binarizovať alebo upraviť línie do odtieňov šedej. Nepoužiteľná oblasť je taká kde nie je možné rozoznať žiadne papilárne línie. Takéto oblasti je vhodné určiť a nepoužiť ich ďalej v spracovaní. [23]

### 3.1 Normalizácia

Normalizácia, mení hodnotu pixelu podľa jeho starej hodnoty a určitých globálnych parametrov. Čiže sa nezaobera susednými pixelmi. Takáto metóda nie je teda samá o sebe postačujúca nakoľko nedokáže vylepšiť drobné chyby v papilárnych líniách. Je ale vhodnou metódou ako prvá úprava obrázku, ktorý bude následne postupovať na ďalšie spracovanie. Nová hodnota pixelu je určená podľa nasledujúceho vzorca 3.1. [23]

$$P'[x, y] = \begin{cases} m_0 + \sqrt{(P[x, y] - m)^2 \cdot v_0/v} & \text{pre } P[x, y] < m \\ m_0 - \sqrt{(P[x, y] - m)^2 \cdot v_0/v} & \text{pre } P[x, y] \geq m \end{cases} \quad (3.1)$$

Kde  $m$  určuje priemernú hodnotu pixelu celého obrázku a  $v$  určuje odchýlku,  $m_0$  a  $v_0$  potom určujú požadované hodnoty priemeru a odchýlky. Takáto metóda sa dá použiť aj blokovej forme, čiže budeme uvažovať priemer a odchýlku v určitej oblasti a však ani takáto modifikácia nie je schopná opraviť nedostatky papilárnych línií [14].



Obrázok 3.2: Ukážka normalizácie obrázku, prevzaté z [14].

### 3.2 Kontextové filtre

Kontextové filtre patria medzi najpoužívanejšie. Namiesto toho aby sa použil jeden filter na celý obrázok, je použitých niekoľko filtrov v závislosti od danej oblasti a kontextu. Kontextom sa myslia orientácie a frekvencie línií v danej oblasti. Zvyčajne je pred-vypočítaných niekoľko filtrov a následne je použitý jeden z nich, tak aby zachoval štruktúru línií a obmedzil šum v obrázku. Existuje niekoľko kontextových filtrov, ale majú rovnaké ciele a to spájanie malých medzier a vyplňanie miest spôsobené šumom alebo pórmi v pokožke. Druhý prístup filtrovania je zväčšenie medzier medzi líniami a oddelenie spojených paralelných línií [23].

#### 3.2.1 O’Gorman a Nickerson filter

Jedna z prvých metód, ktorá používala kontextové filtre na zlepšenie kvality obrázku bola uvedená v roku 1988 [23]. Táto metóda používa jeden hlavný filter, ktorý závisí na štyroch parametroch a to: minimálna a maximálna šírka papilárnej línie, minimálna a maximálna



šírka ryhy. Po vytvorení hlavného filtra, sa vygeneruje 16 ďalších filtrov, každý s jemným pootočením o  $22,5^\circ$ . Následne prebieha samotné filtrovanie, na základe natočenia línií v obrázku sa vyberie a použije vhodný filter [23].

### 3.2.2 Gaborov filter

Gaborov filter používa dva spôsoby výberu vhodného filtra a to frekvencia línií a natočenie línií. Využíva fakt, že priebeh tvaru a rýh sa v smere línie mení. Použitím vhodných filtrov teda môže efektívne rekonštruovať zašumené a poškodené časti odtlačku [14].

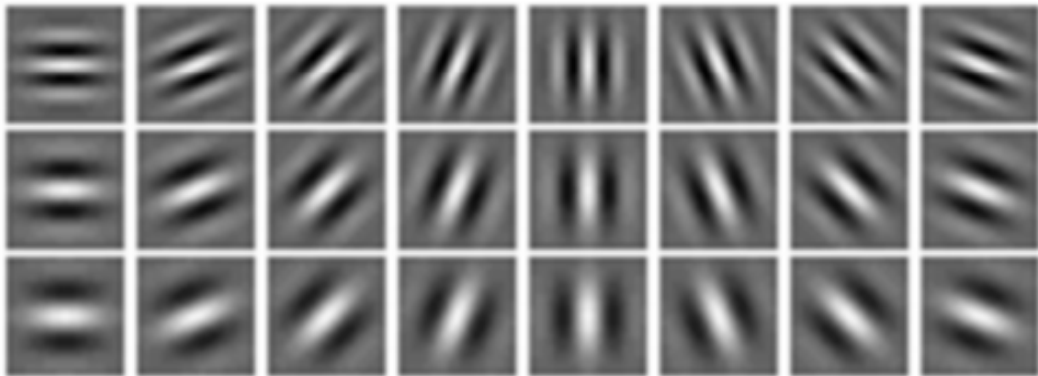
Aby mohol byť použitý takýto filter, potrebujeme špecifikovať štyri parametre:  $f, \theta, \sigma_x, \sigma_y$ . Frekvencia  $f$  a natočenie  $\theta$  je daná konkrétnou oblasťou, ktorá sa bude filtrovať. Ďalšie dva parametre sú nastaviteľné ale zároveň predstavujú kompromis medzi nízkou zlepšenou kvalitou alebo chybnou rekonštruovanou oblasťou. Čím väčšie hodnoty nastavíme, tým lepšie budú redukovať šum a vyplňať prípadné chyby ale zároveň môže prísť k nesprávnej rekonštrukcii. Čím menšie hodnoty nastavíme tým menšie bude riziko že do obrázka zavedieme chyby, avšak za cenu menšej redukcie šumu a opravy. Určenie správnych hodnôt  $\sigma_x$  a  $\sigma_y$  je teda kľúčové. Experimentami je možné zistiť vhodné hodnoty. Pre rýchlosť výpočtu sú doporučené hodnoty  $\sigma_x = \sigma_y = 4$ . Nasledujúci vzťah definuje Gaborov filter:

$$g(x, y, \theta, f) = \exp\left\{-\frac{1}{2} \left[ \frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2} \right]\right\} \cdot \cos(2\pi f \cdot x_\theta) \quad (3.2)$$

$$x_\theta = x \cdot \cos \theta + y \cdot \sin \theta \quad (3.3)$$

$$y_\theta = -x \cdot \sin \theta + y \cdot \cos \theta \quad (3.4)$$

V rovniciach 3.2, 3.3 a 3.4  $\sigma_x$  a  $\sigma_y$  predstavujú odchýlky osy  $x$  a  $y$ ,  $x_\theta$  a  $y_\theta$  sú súradnice bodov po natočení uhlom  $\theta$ . [23]



Obrázok 3.3: Súbor vygenerovaných filtrov, z ktorých bude vybrané pri filtrovaní.

## 3.3 Ochorenie pokožky

Medzi ochorenia, ktoré postihujú pokožku v oblasti prstov patria napríklad bradavice, psoriáza či ekzémy. Vírus spôsobujúci bradavice sa nazýva papillomavírus. Existuje viac ako sto

druhov tohoto vírusu. Konkrétne druhy vírusu spôsobujú bradavice na špecifických miestach. Infekcia týmto vírusom je pomerne častá a týka sa až 10 % populácie. Dôležitá je včasná liečba bradavice kvôli zabráneniu šírenia na iné časti pokožky. Liečba pozostáva z odstránenia bradavice a posilnenie imunitného systému. Medzi možné liečby patrí odstránenie laserom alebo kvapalným dusíkom [32]. Bradavice predstavujú problém v súvislosti odtlačkov prstov, pretože sa často nachádzajú v miestach papilárnych línií a znehodnocujú tak kvalitu samotného odtlačku prstu. Nasnímané odtlačky, ktoré sú postihnuté týmto ochorením sú znázornené na obrázku 3.4. [17]



Obrázok 3.4: Odtlačky poškodené bradavicami, prevzaté z [17].

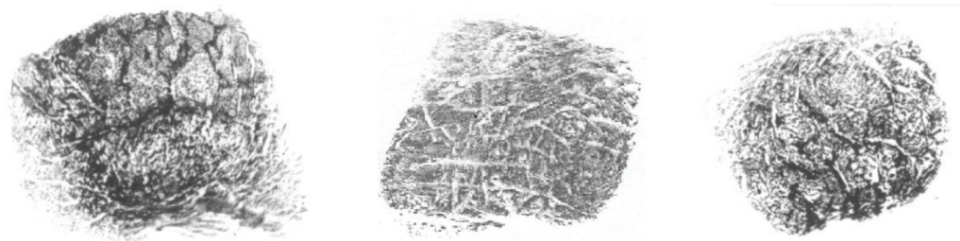
Ekzém je neinfekčné zápalové ochorenie postihujúce pokožku. Medzi prejavy patria výskyt svrbivých červených oblastí, ktoré neskôr mokvajú a vytvárajú pluzgiere. Pri tomto ochorení môžu nastávať aj zmeny farby pokožky [5]. Toto ochorenie postihuje 15 až 25 % populácie, z toho je 20 % detí. Existuje celkovo 5 druhov ekzémov. Medzi príčiny vzniku patrí aj alergická reakcia. Ekzém môže postihovať celú ruku vrátane prstov [17]. Z hľadiska identifikácie osoby pomocou odtlačku prstu vzniká týmto ochorením problém. Ukážka nasnímaných odtlačkov prstov s ekzémami je na obrázku 3.5.



Obrázok 3.5: Odtlačky poškodené ekzémami, prevzaté z [5].

Medzi ďalšie ochorenia, ktoré priamo ovplyvňujú kvalitu odtlačku prstu sa radí psoriáza. Nasnímané odtlačky prstov zasiahnuté psoriázou ukazuje obrázok 3.6. Toto ochorenie je chronické a tak isto ako ekzém, je zápalového typu. Prejavuje sa ako šupinaté alebo začervenané ložiská na pokožke. Popraskaná pokožka môže v týchto ložiskách krváčať. Kvôli týmto príznakom je často krát na nerozoznanie od ekzému. Charakteristické je striedanie období akútneho zápalu a období s miernymi alebo žiadnymi príznakmi. Ochorenie postihuje

huje 1 až 3 % celkovej populácie. Existuje mnoho ďalších ochorení, ktoré postihujú oblasti prstov. [5]



Obrázok 3.6: Odtlačky poškodené psoriázou, prevzaté z [5].

## 3.4 Kvalita a verifikácia

Medzi možné metriky, ktoré môžeme v odtlačkoch merať patria napríklad kvalita odtlačku a hodnota verifikácie. Nástroje, ktoré umožňujú merať kvalitu sú napríklad NFIQ2 a FiQ<sub>i</sub>Vi. Na verifikáciu odtlačkov bude použitý v tejto práci nástroj VeriFinger.

### 3.4.1 NFIQ2

Výsledky biometrických systémov sú závislé na kvalite vzoriek. Ak nejakým spôsobom zlepšime kvalitu obrázku, či už dizajnom senzora alebo dodržiavaním štandardov, získame tak lepši výkon takéhoto systému. Ak zistíme u odobratej vzorky že kvalita je nedostatočná, môžeme vzorku odobrať znovu, použiť inú metódu snímania odtlačku alebo v reálnom čase vybrať najlepšiu odobratú vzorku a uložiť ju pre ďalšie použitie. Takýto algoritmus môže byť jednoducho automatizovaný a zaradený do biometrického systému za účelom zvýšenia jeho výkonu. Jedným z nástrojov, ktoré ponúkajú takúto analýzu je NFIQ2 (NIST Fingerprint Image Quality) <sup>1</sup>. Pôvodný účel tohoto nástroja bola predpoveď výkonu pre programy, ktoré porovnávajú odtlačky, tým že napríklad meria rýchlosť výpočtu polohy markantov na odtlačku. Výrobci zariadení pre snímanie odtlačkov tak mohli nastavovať parametre ich snímačov, tak aby rýchlosť a kvalita bola optimálna. Výstup nástroja NFIQ2 je číslo v rozmedzí 0 až 100 bodov, podľa štandardu kvality biometrickej vzorky. [16]

### 3.4.2 VeriFinger

VeriFinger je program, ktorý poskytuje identifikačné funkcie pre biometrické systémy. Je založený na hlbokých neurónových sieťach. Pre identifikáciu používa markanty ale aj niekoľko ďalších algoritmov pre zvýšenie výkonu porovnávania odtlačkov. Tento program dokáže vzájomne porovnávať aj rolované odtlačky s vysokou presnosťou a spoľahlivosťou vďaka tomu že je tolerantný voči deformáciám. Taktiež je tolerantný aj voči rotáciám a transláciám. Dokáže identifikovať aj ťažko poškodené odtlačky iba na základe piatich až siedmich markantov. Rýchlosť porovnávania môže byť veľmi vysoká, až 40 000 odtlačkov za sekundu. Poskytuje zároveň identifikáciu (porovnanie odtlačku 1:N) a aj verifikáciu (porovnanie odtlačku 1:1). Ak je obrázok poškodený, adaptívny filtrovací algoritmus eliminuje šum a zlepšuje kvalitu odtlačku na úrovni línií. Jeden odtlačok môže byť nasnímaný viac krát, pričom

<sup>1</sup><https://github.com/usnistgov/NFIQ2>

z každého odtlačku sa vyberú rôzne vzory. Následne sa získané vzory zjednotia a vytvorí sa väčšia množina vzorov, ktorá je uložená do databáze za účelom spoľahlivejšej identifikácie vlastníka odtlačku. [25]

### 3.4.3 IDKit Fingerprint

IDKit Fingerprint SDK (Software development kit) je súbor nástrojov, ktorý umožňuje vývoj aplikácií súvisiacich s odtlačkami prstov. Ponúka rýchlu verifikáciu, identifikáciu a segmentáciu odtlačkov. Medzi jeho výhody patria rýchlosť, presnosť a výkon. Obsahuje taktiež meranie kvality odtlačku, ktorá je počítaná pred každou verifikáciou alebo identifikáciou. V prípade potreby dokáže odstraňovať šum z pozadia odtlačku. Jeho použitie spočíva v priamom volaní funkcií z poskytnutých knižníc. Súčasťou ponúkaných nástrojov je aj nepretržitá podpora školeného tímu. [15]

### 3.4.4 FiQiVi

FiQiVi (Fingerprint quality visualiser) je program pre určenie a vizualizáciu kvality odtlačku prstu. Využíva globálne a lokálne vlastnosti odtlačku prstu pre určenie kvality. Poskytuje vizualizáciu kvality pomocou mapy, jednotlivé oblasti určujú kvalitatívne hodnotenia. Medzi hodnotené vlastnosti patria napríklad prelínanie línií a rýh, pomer línií a rýh či kontrast. Obsahuje taktiež hodnotenie kvality pomocou NFIQ1. [19]

# Kapitola 4

## Neurónové siete

V tejto kapitole bude popísané strojové učenie a úlohy, ktoré sa týkajú tejto oblasti. Po popise biologického a umelého neurónu bude nasledovať úvod do problematiky neurónových sietí. Medzi špecifický druh neurónovej siete patrí aj konvolučná neurónová sieť, ktorá nájde uplatnenie hlavne v spracovaní obrazových dát a bude použitá v tejto práci.

### 4.1 Strojové učenie

Niektoré úlohy, na pohľad veľmi jednoduché a intuitívne pre človeka, môžu byť až príliš ťažko algoritmovateľné. Preto bolo nutné zaviesť nový druh riešenia úloh a to strojové učenie. Kedy sa program naučí ako riešiť konkrétnu úlohu. Princíp strojového učenia spočíva typicky v transformovaní  $n$ -rozmerného vstupného vektora, ktorý reprezentuje kolekciu vlastností, na výstupný vektor. [9]

Medzi typické úlohy patria:

- klasifikácia - pri tomto druhu úlohy sa program snaží priradiť vstupu jednu z  $k$  kategórií, teda vytvára funkciu  $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$ . Medzi typické príklady patrí napríklad rozpoznávanie objektov na obrázku.
- Regresia - v tomto type úlohy program predpovedá výstup na základe vstupu, teda vytvára funkciu:  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , ako príklad môže byť predpoveď vývoja trhu s akciami.
- Štruktúrovaný výstup - predstavuje ľubovoľnú úlohu, ktorej výstupom je vektor, pričom sa kladie dôraz na vzťahy medzi jednotlivými elementami tohoto vektora. Napríklad vytvorenie vety popisujúci obrázok je zaradené do tejto kategórie.
- Syntéza - tento druh úlohy spočíva v generovaní nových výstupov, ktoré sú podobné vstupným. Takéto využitie je možné uplatniť napríklad v generovaní textúr kedy dizajner nemusí riešiť každý pixel ale namiesto toho využije generovanie veľmi podobného vzoru.
- Odstraňovanie šumu - takýto algoritmus dostáva na poškodený vstup, z ktorého sa potom snaží dostať pôvodný nepoškodený. Použije je vhodné napríklad pri odstraňovaní šumu v obrázku.

Vyhodnotenie modelu môžeme určiť ako presnosť, ktorá koreluje s počtom správne určených vstupom alebo použijeme chybovosť, ktorú zase určuje počet nesprávne určených

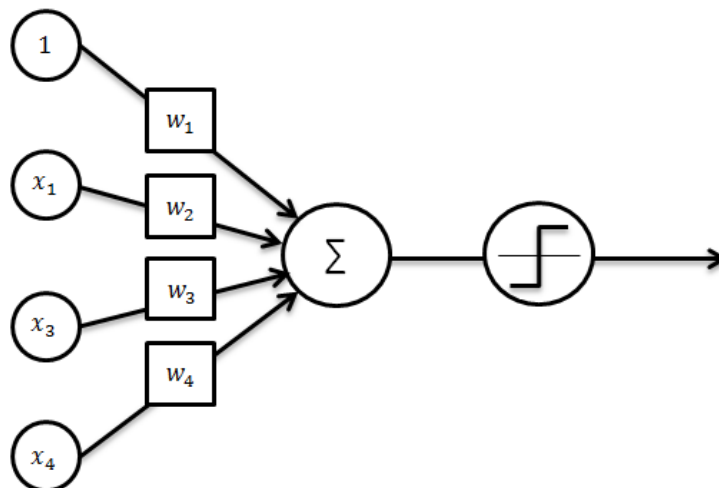
vstupov. Toto učenie môžeme rozdeliť na dva druhy a to *učenie bez učiteľa* a *učenie s učiteľom* [9]. Učenie s učiteľom používa dataset, ktorý obsahuje vstup pre model a zároveň aj chcený výstup. Model sa potom snaží minimalizovať chybu svojho výstupu podľa tohoto datasetu. Pri učení bez učiteľa dataset neobsahuje chcený výstup pre každý vstup a učí sa na základe informácií, ktoré si sám vytvára. Určitá forma odmeny potom koriguje učenie takéhoto modelu. [9]

## 4.2 Biologický neurón

Typický biologický neurón je zložený z bunkového tela, dendritov a axónov. Telo bunky má na starosti látkové premeny. Dendrity prijímajú vstupné vzruchy a axóny prenášajú vzruchy neurónu do ďalších neurónov. Medzi jednotlivými neurónmi prebieha komunikácia pomocou synapsií. Typický neurón má jeden axón a niekoľko rozvetvených dendritov, na ktorých sú synapse. Funkcia neurónu spočíva v získavaní, prenose, spracovaní a ukladaní informácie. Prenos informácií je zabezpečený elektrickými signálmi [20].

## 4.3 Umelý neurón

Umelý neurón je štruktúrovaný podobne ako biologický neurón. Základnú schému popisuje obrázok 4.1. Má niekoľko vstupov, ku každému vstupu je priradená váha. Všetky vstupy sú vynásobené príslušnou váhou a následne sčítané. Aktivačná funkcia rozhodne či neurón pošle signál ďalej, prípadne ho modifikuje. [26]



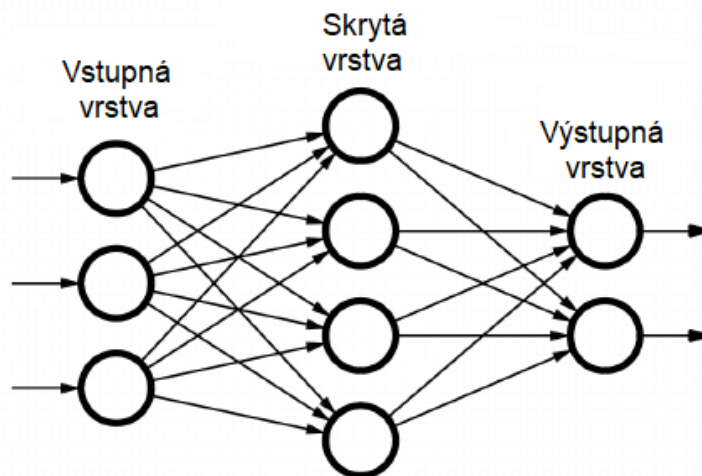
Obrázok 4.1: Model umelého neurónu.

## 4.4 Umelá neurónová sieť

Umelá neurónová sieť je súbor prepojených neurónov, ktorých funkcionalita je založená na fungovaní biologického neurónu. Jej schopnosť spracovávať dáta je uložená v prepojeniach

medzi jednotlivými neurónmi v tzv. váhach, získaných adaptáciou na učenie siete. Dopredná neurónová sieť sa skladá z neurónov usporiadaných do takej topológie, kde neuróny nespolupracujú s predchádzajúcimi vrstvami. Iným typom siete môže byť rekurentná neurónová sieť kedy neuróny môžu spolupracovať aj s nižšími, predchádzajúcimi vrstvami. [11]

Neuróny v sieti sú štruktúrované do vrstiev. Vrstva neurónov, do ktorej smeruje vstupný vektor sa nazýva vstupná vrstva. Ďalej nasleduje skrytá vrstva, ktorá nie je vstupnou ani výstupnou. Neurónová sieť môže obsahovať viacero skrytých vrstiev. Jednovrstvová sieť obsahuje jednu skrytú vrstvu. Viac vrstvová sieť sa skladá aspoň z dvoch skrytých vrstiev. Posledná výstupná vrstva poskytuje výstup siete. Sieť, ktorá obsahuje aspoň jednu skrytú vrstvu, sa nazýva hlboká neurónová sieť. Podľa prepojenia neurónov v rámci vrstiev môžeme rozlišovať taktiež druh vrstvy. Základným typom vrstvy je plne prepojená vrstva, v ktorej každý neurón je prepojený s každým neurónom v nasledujúcej vrstve [26].



Obrázok 4.2: Model umelej neurónovej siete.

Obrázok 4.2 popisuje jednoduchú doprednú umelú neurónovú sieť. Obsahuje jednu skrytú, plne prepojenú vrstvu so štyrmi neurónmi a jednu výstupnú vrstvu s dvomi neurónmi.

## 4.5 Konvolučná neurónová sieť

Druhom neurónovej siete je aj konvolučná neurónová sieť. V týchto sieťach sú taktiež neuróny uložené vo vrstvách a predmetom učenia je zmena váh prepojení neurónov. Hlavným rozdielom medzi konvolučnou a klasickou neurónovou sieťou je organizácia neurónov v jednotlivých vrstvách. V klasickej neurónovej sieti existujú prepojenia neurónov také že každý neurón je prepojený s každým neurónom z predchádzajúcej vrstvy. Tým pádom máme v sieti veľké množstvo prepojení, a teda aj veľké množstvo potrebných výpočtov. Keďže sa tieto siete zameriavajú na spracovanie obrazových informácií, množstvo neurónov a parametrov, ktoré sieť musí obsahovať je veľmi veľké. Farebný obrázok môže obsahovať niekoľko kanálov pre jednotlivé farby, týmto sa parametre zase navyšujú. V konvolučnej vrstve je však neurón napojený iba na určitú časť obrázku. Týmto sa redukuje počet potrebných prepojení a parametrov. Veľkosť oblasti, ktorú daný neurón spracováva sa nazýva *receptive field size* [23]. Medzi ďalšiu podstatnú vlastnosť konvolučných neurónových sietí patrí

zdieľanie parametrov siete. Týmto spôsobom sa znižujú nároky na pamäť potrebnú pre spracovanie vstupu. Predpokladá sa že vstupy obsahujú niektoré podobné rysy, ktoré sú reprezentované týmito zdieľanými parametrami. Výstupom neurónov je aktivačná mapa, ktorá predstavuje niektorý z rysov vstupu. Aktivačná mapa je potom priamo závislá na daných zdieľaných parametroch. Týchto aktivačných máp môže byť na výstupe niekoľko a teda detegujú niekoľko rysov v danom vstupe [23].

#### 4.5.1 Konvolúcia

Je matematická operácia, ktorá má na vstupe dve funkcie, a výsledkom je tretia funkcia, ktorá popisuje ako ovplyvňuje jedna funkcia druhú [31]. V konvolučných neurónových sieťach sa používa diskretná verzia konvolúcie a má nasledovný tvar:

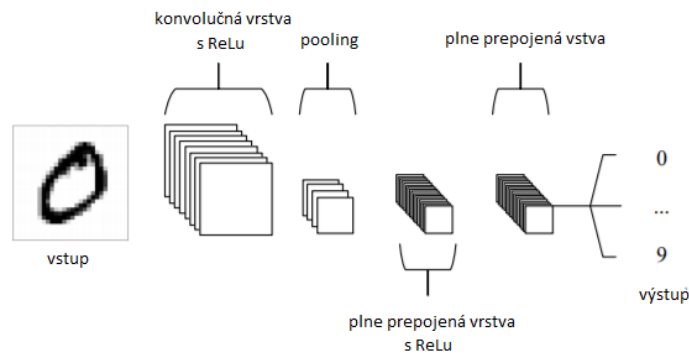
$$(f * h)(x, y) = \sum_i \sum_j f(x - y, y - j) \cdot h(i, j) \quad (4.1)$$

Kde funkcia  $h$  predstavuje kernel - filter, a funkcia  $f$  predstavuje obrázok. Ak by sme napríklad použili ako filter maticu 3x3, kde každá bunka matice bude mať hodnotu 1/9, na výstupe by sme dostali rozostrený obrázok. Podobne môžeme použiť napríklad filter na detekciu hrán. Takéto filtre sú kľúčové v konvolučných sieťach.

#### 4.5.2 Architektúra

Konvolučné neurónové siete sa skladajú z troch typov vrstiev. Medzi ne patria: konvolučná vrstva, tzv. *pooling* vrstva a plne prepojená vrstva. [27]

Obrázok 4.3 ilustruje jednoduchú konvolučnú neurónovú sieť schopnú klasifikovať cifry z obrázku.



Obrázok 4.3: Jednoduchá architektúra konvolučnej siete, prevzaté a upravené z [27].

Túto jednoduchú architektúru môžeme teda rozdeliť na niekoľko častí:

- Konvolučná vrstva - predstavuje vrstvu, ktorá počíta konvolúciu. Obsahuje ľubovoľné množstvo filtrov. Úlohou tejto vrstvy je detekcia vlastností vstupu. Napríklad detekcia hrán v obrázku, pričom každý filter deteguje iné vlastnosti obrázku.
- Pooling vrstva - úlohou tejto vrstvy je zmenšenie rozmerov jej vstupu, použitím filtra s rôznou veľkosťou a krokom. Filter môže napríklad vyberať najväčšiu hodnotu, alebo



vypočíta priemernú hodnotu z pixelov. Niekedy sa nazýva aj *downsampling* vrstva. Podobne pracuje aj *up sampling* vrstva avšak z malého obrázku vytvára väčší.

- Plne prepojená vrstva - Každý neurón v tejto vrstve je prepojený na každý neurón v predchádzajúcej vrstve, ale jednotlivé neuróny nie sú prepojené so sebou v rámci vrstvy. Takéto vrstvy sa používajú v klasických neurónových sieťach.
- Aktivačná funkcia - ma za úlohu riadiť výstupy jednotlivých neurónov. Na základe vstupov konkrétneho neurónu, potom použije aktivačnú funkciu a určí hodnotu jeho výstupu. Medzi typické aktivačné funkcie patrí ReLu, ktorá nezáporný výstup necháva bez zmeny a záporný zmení na nulu. [27]

## 4.6 Postupy učenia siete

Pre pochopenie teórie učenia siete je nutné vysvetliť niekoľko termínov a techník. Nasledujúca sekcia popisuje hodnotenie výstupov siete pomocou *loss* funkcie, techniky rozdeľovania vstupnej databázy, ktorá bude použitá pre učenie a vyhodnocovanie výstupov. Ďalšími fundamentálnymi problémami z oblasti neurónových sietí sú *overfitting* a *underfitting*, ktoré budú taktiež vysvetlené.

### 4.6.1 Loss funkcia

Ak chceme sledovať kvalitu výstupu neurónovej siete, musíme určiť mieru, ktorá ju popisuje. Táto miera sa nazýva loss funkcia. Zahŕňa výpočet odlišnosti očakávaného výstupu a aktuálneho výstupu siete. Loss hodnota je vyhodnocovaná po každej epoche. Epocha je jeden tréningový cyklus, počas ktorej sú na vstup siete vkladané jednotlivé dávky [27]. Dávka je počet tréningových vstupov siete, po ktorom nasleduje úprava váh v sieti. Výber veľkosti dávky je veľmi dôležitý pri práci učení siete. Príliš malá alebo veľká dávka môže viesť k neúspešnému učeniu siete [29]. Táto funkcia je použitá ako spätná väzba pre optimalizáciu siete, čiže nastavovanie váh, o ktoré sa stará optimalizátor. Knižnice pre vytváranie neurónových sietí ponúkajú na výber niekoľko optimalizátorov. [11] [27] [3]

### 4.6.2 Rozdelenie datasetu

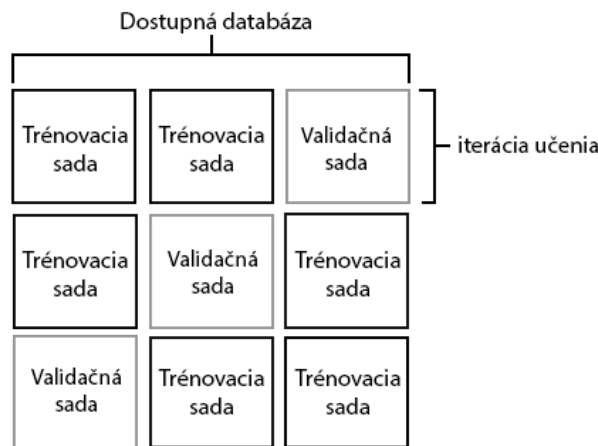
Pre učenie takéhoto systému potrebujeme vstupné dáta - dataset. Tie následne rozdelíme na tréningovú a validačnú sadu. Učenie bude prebiehať iba na tréningovej sade. Úspešnosť systému potom vyhodnotíme na validačnej sade. Cieľom takéhoto postupu je pripraviť systém na vstupy, ktoré ešte nikdy nedostal. Ďalšou sadou bude testovacia sada, tá obsahuje vzorky, ktoré systém vôbec nevidel a budú na nej vykonávané testy v rámci úspešnosti rekonštrukcie. Existuje niekoľko možností, ktorými možno rozdeľovať dataset. Dôležitým aspektom pri výbere techniky rozdelenia datasetu je jeho veľkosť. Patria sem napríklad:

- *hold out* - dataset rozdelíme na dve časti, tréningovú a validačnú v určitom pomere, napríklad 80 % tréningových dát a 20 % validačných dát, ako je vidieť na obrázku 4.4. Tento spôsob je najjednoduchší ale zároveň nemusí byť vždy najvhodnejší. Problémom použitia tohoto rozdelenia môže nastať v prípade že máme nedostatočnú veľkosť celkového datasetu a sieť sa tak nebude môcť naučiť všetky potrebné vzory. Túto chybu môžeme odhaliť tak že vzorky pred učením náhodne zamiešame, ak následne dostaneme po každom novom zamiešaní významne rozdielne výsledky, vieme že veľkosť tréningovej sady je nedostatočná,



Obrázok 4.4: Rozdelenie dostupného datasetu na tréningovú a validačnú množinu.

- *k-fold* - tento spôsob rozdelenenia datasetu je založený na rozdelení celej sady na  $k$  častí, pričom jedna časť slúži na validáciu a zvyšných  $k-1$  častí slúži na tréning siete, výsledné hodnoty loss funkcie potom predstavuje priemer nameraných hodnôt v jednotlivých učeniach. Obrázok 4.5 znázorňuje schému rozdelenia *k-fold*,



Obrázok 4.5: Rozdelenie dostupného datasetu na tréningovú a validačnú množinu.

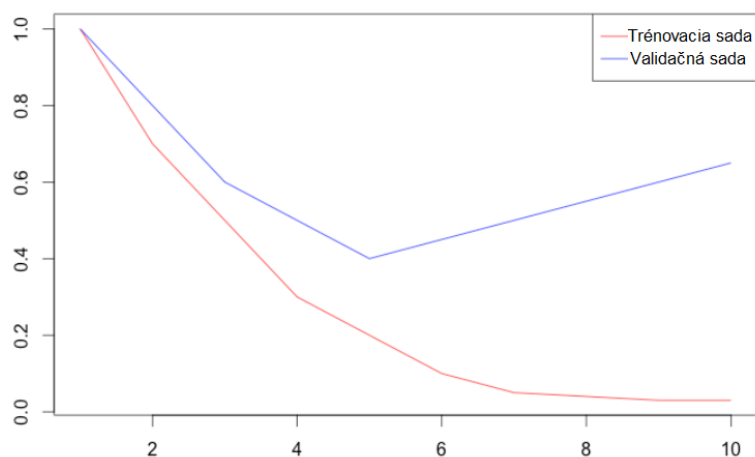
- *k-fold* s náhodným zamiešaním - tento spôsob je vhodné použiť pokiaľ je dataset veľmi malý a model potrebujeme čo najpresnejšie vyhodnotiť, je založený na rozdelení datasetu na  $k$  častí, avšak pred každou iteráciou učenia náhodne zamiešame vzorky datasetu, dostaneme tak unikátne množiny v každej iterácii, pomocou ktorých bude možné výsledný model ohodnotiť presnejšie. [3] [27]

### 4.6.3 Overfitting a underfitting

*Overfitting* alebo *underfitting* vzniká pri riešení každej úlohy z oblasti strojového učenia a je jedným z hlavných problémov pri vytváraní neurónovej siete. Učením siete chceme dosiahnuť zároveň čo najlepšiu optimalizáciu a zároveň generalizáciu. Optimalizácia je proces úpravy modelu siete kedy sa snažíme aby model dosiahol čo najlepšie výsledky na tréningových dátach. Generalizácia je miera, ktorá určuje ako dobre model spracováva vstupy, ktoré ešte nikdy nevidel, čiže validačné dáta. Cieľom učenia siete čo najlepšia generalizácia. Na začiatku učenia, optimalizácia a generalizácia spolu korelujú. Čím menšia je chyba výstupu s použitím tréningových dát, tým menšia je chyba validačných dát. Táto fáza učenia sa nazýva *underfitting*, sieť zatiaľ nedosiahla svoj najlepší potenciál a potrebuje niekoľko ďalších epoch lepšie spracovanie vzorov v dátach. Po tejto fáze učenia nastáva *overfitting*. Chyba výstupov pre tréningové dáta sa znižuje avšak, chyba výstupov pre validačné dáta sa zvyšuje, a generalizácia sa začne zhoršovať. Sieť sa začína učiť vzory, ktoré sú typické pre tréningové

dáta ale zároveň nie sú obsiahnuté vo validačných dátach. Obrázok 4.6 znázorňuje *overfitting*. Aby sme predišli tomu že model sa naučí spracovávať tieto zavádzajúce vzory, musíme urobiť určité opatrenia. Najlepším je získanie väčšieho množstva tréningových dát. Model, ktorému poskytneme väčšiu dátovú sadu bude prirodzene lepšie generalizovať. V prípade že zdroje tréningových dát sú limitované, ďalším opatrením je obmedzenie počtu vzorov, ktoré dovolíme sieti naučiť. Pokiaľ si sieť bude môcť zapamätať iba určitý počet vzorov, prinúti ju to zamerať sa iba na tie najdôležitejšie. Proces, ktorý zabraňuje *overfittingu* sa nazýva regularizácia a zahŕňa niekoľko techník, patria medzi ne napríklad:

- Zmenšenie počtu alebo veľkosti vrstiev - je jedným z jednoduchých opatrení, ktoré môžeme použiť pre redukovanie *overfittingu* siete, ktorá bude mať limitovanú kapacitu, vzorov, ktoré si má zapamätať, sa zameria na tie podstatné na rozdiel od tých, ktoré sú obsiahnuté prevažne vo validačnej sade, týmto donútíme sieť lepšie generalizovať, treba však brať ohľad taktiež na dostatočnú kapacitu siete takú aby dokázala obsiahnuť všetky potrebné vzory, ladenie vrstiev je teda vyvažovanie týchto dvoch aspektov, postup pri určovaní počtu vrstiev by mal teda začínať pri malých modeloch s postupným pridávaním a zväčšovaním vrstiev.
- *Dropout* vrstvy - tieto vrstvy majú za úlohu vynechávať náhodne hodnoty vstupu, týmto spôsobom zabránime sieti aby sa naučila nepodstatné informácie tým pádom ovplyvňujeme priebeh *overfittingu*, pre príklad uvažujme na vstupe tejto vrstvy vektor  $v = [2, 3, 5, 6, 7]$ , jej výstupom bude teda vektor, v ktorom budú náhodné hodnoty nastavené na nulu, v tomto prípade bude výstupom vrstvy vektor s hodnotami napríklad  $v' = [2, 0, 5, 0, 7]$ , pri ladení siete určujeme hodnotu udávajúci časť výstupu, ktorá bude vynechaná.
- Regularizácia váh - jednoduchší model, má menšiu tendenciu pre vznik *overfittingu*, tento fakt využíva aj technika, ktorá sa nazýva regularizácia váh, ide o cieleňú zmenu parametrov pri optimalizovaní siete, regulovaním váh môžeme predísť *overfittingu*. [3]



Obrázok 4.6: Ukážka *overfittingu*, ktorý nastáva zhruba v piatej epoche, prevzaté a upravené z [22].

## Kapitola 5

# Generovanie syntetických odtlačkov prstov a simulácia poškodenia

S narastajúcou popularitou odtlačkov prstov v biometrii, rastie aj dopyt po odtlačkoch, na ktorých bude možné testovať a vyvíjať softvér potrebný pre rôzne disciplíny v tejto oblasti. S reálnymi odtlačkami zozbieranými od ľudí sú spájané mnohé problémy. Jedným z nich je cena odtlačku či už časová alebo finančná, je príliš vysoká na to aby bolo možné efektívne zbierať takéto dáta. Ďalším problémom je bezpečnosť, človek ktorý daruje jeho odtlačok podstupuje riziko úniku dát, ktoré nie každý je ochotný podstúpiť. Preto tieto a mnohé iné problémy je tu možnosť generovania syntetických odtlačkov prstov. Syntetické odtlačky prstov predstavujú vygenerované odtlačky podobné ľudskému prstu. Jedným z programov na generovanie takýchto odtlačkov je SFinGe [23]. Tento program dokáže generovať rozsiahle databáze odtlačkov a tak poskytovať potrebné dáta pre učenie a testovanie algoritmov. Odtlačky vytvorené týmto programom budú použité aj v tejto práci. Program SFinGe dokáže simulovať obrázky získané z elektronických snímačov prstov vrátane šumu. Pre simuláciu poškodenia budú využité výstupy z nástroja SyFDaS [18].

### 5.1 SFinGe

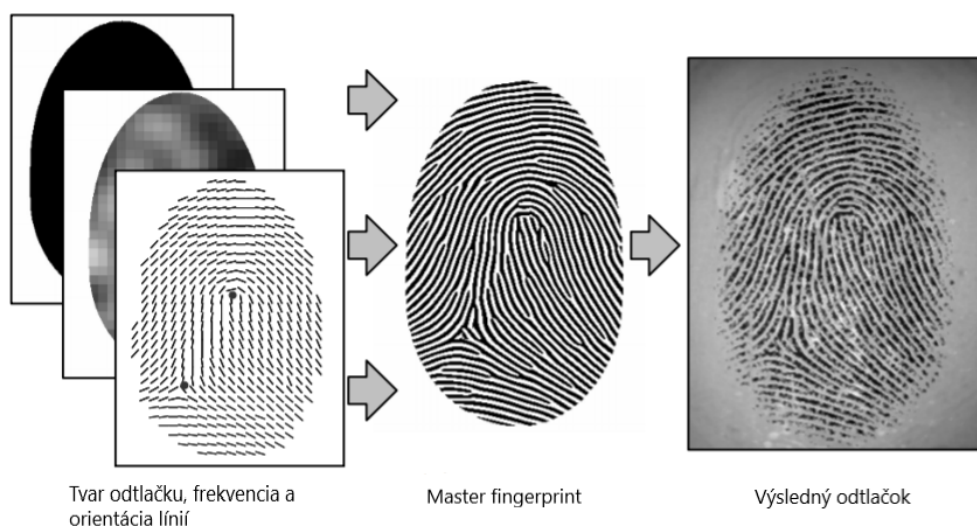
SFinGe je program, pomocou ktorého je možné generovať syntetické odtlačky. Generovanie odtlačku prebieha iteratívnym rozširovaním jednej alebo niekoľkých počiatočných oblastí. Po finálnom zašumení a vyrenderovaní odtlačku vzniká veľmi realistický odtlačok. Generovanie prebieha v niekoľkých krokoch:

1. Vytvorenie počiatočných jadier a delt,
2. generovanie pola orientácií,
3. generovanie mapy hustoty línií,
4. generovanie vzorov línií,
5. vkladanie šumu a renderovanie.

Ponúka nastavenie množstva vlastností odtlačkov, vďaka ktorým sú výstupy veľmi variabilné. Generovanie syntetického odtlačku začína definovaním tvaru prsta a ostatných

vlastností, z tých sa následne vytvorí *master fingerprint* odtlačok, z ktorého je možné odvodzovať ďalšie odtlačky rôznymi modifikáciami. Postup generovania odtlačku je znázornený na obrázku 5.1. Medzi možné modifikácie odtlačku patria:

- Tvar odtlačku,
- simulácia snímania iba určitej časti odtlačku,
- rôzne šírky línií (vlhkosť prsta, tlak),
- skreslenie,
- šum,
- malé porezania,
- rotácie a translácie,
- generovanie pozadia.



Obrázok 5.1: Postup generovania syntetického odtlačku, prevzaté z [23].

## 5.2 Anguli

Ďalší program na generovanie odtlačkov je Anguli, vyvinutý Indickým inštitútom vedy, napísaný v jazyku C++. Anguli je voľne dostupný na rozdiel od prechádzajúceho programu. Nesie to so sebou však aj určité nevýhody. Oproti SFinGe je mierne obmedzený, miera nastavenia vzhľadu prsta nie je taká veľká. Anguli neposkytuje väčšinu funkcií, ktorými disponuje SFinGe. Používa taký istý algoritmus ako SFinGe. Avšak nastavenia zahŕňajú iba šum, transláciu, rotáciu a vrásky. Dokáže pomerne rýchlo generovať rozsiahle databázy odtlačkov. Súbor 1 000 000 obrázkov vygeneruje za menej ako 4 dni, pri použití procesoru s 8 jadrami. Obrázok 5.2, ukazuje vygenerované syntetické odtlačky. [13]



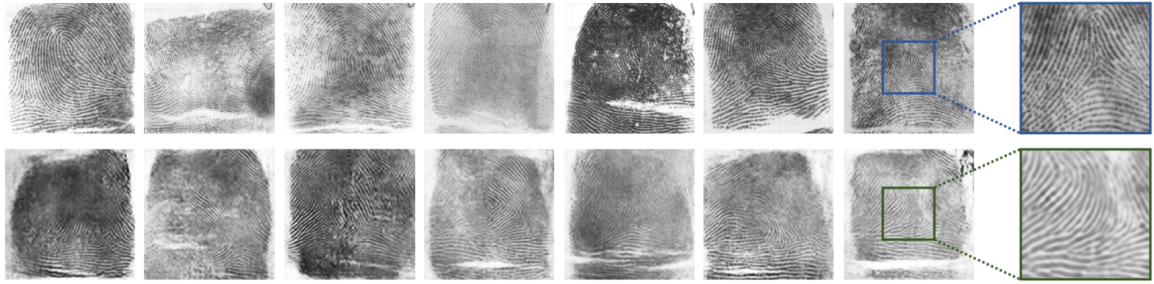
Obrázok 5.2: Syntetické odtlačky, vygenerované pomocou programu Anguli.

### 5.3 SYnFI

Okrem matematických modelov pre vytváranie syntetických odtlačkov, aké používajú Anguli a SFinGe, existujú aj modely založené na neurónových sieťach. Jednou z prác, ktorá využíva neurónové siete je aj *SYnFI: Automatic Synthetic Fingerprint Generation* [28]. Cieľom tejto práce bolo generovanie syntetických odtlačkov prstov, ktoré sú nerozoznateľné od reálnych odtlačkov.

Použitá architektúra je pomerne rozsiahla, skladá sa z GAN (Generative Adversarial Network) a SR (Super resolution) sietí. Pre učenie sietí bola použitá databáza odtlačkov *NIST Special Database 9*, obsahujúca celkovo 54 000 odtlačkov. Čiernobiele odtlačky boli uložené vo 8-bitovom formáte `png`. Tieto odtlačky obsahujú taktiež pohlavie subjektu a klasifikáciu prstu. Učenie týchto dvoch sietí prebiehalo na serveri s pomerne veľkým výpočtovým výkonom. Server bol vybavený so 128GB pamäte RAM, dvoma 12-jadrovými procesormi a štyrmi grafickými kartami NVIDIA Titan XP. Generovanie prebieha v troch fázach, prvou je príprava databázy odtlačkov do vhodnej formy. Keďže vzorky z použitej databázy obsahovali okrem odtlačku aj zložky ako čísla a iné kódy, bolo treba vzorky upraviť. Táto úprava zahŕňala zistenie hraníc odtlačku a následné vystrihnutie časti odtlačku z obrázka. Z týchto upravených vzoriek bolo treba vytvoriť dve sady. Prvá sada odtlačkov bola použitá pre sieť GAN, a vzorky tejto sady mali rozlíšenie  $64 \times 64$  pixelov. Druhá sada bola použitá ako vstup pre sieť SR, a vzorky mali rozlíšenie  $256 \times 256$  pixelov. [28]

V druhej fáze procesu sú generované syntetické odtlačky špecifickou sieťou WGAN (Wasserstein GAN). Pomocou upravenej databázy je sieť naučená pre vytváranie realistických odtlačkov. Avšak sieť dokáže vytvárať kvalitné obrázky iba o veľkosti  $64 \times 64$  pixelov. V poslednej fáze, je potrebné pre vytvorenie odtlačkov väčšieho rozlíšenia, upraviť obrázky ďalšou neurónovou sieťou, ktorá zvyšuje rozlíšenie, a tou je ESRGAN (Enhanced Super-Resolution Generative Adversarial Networks). Táto sieť pozostáva z niekoľkých konvolučných vrstiev a viacerých plne prepojených vrstiev, ktoré sú popísané v podkapitole 4.5. Po vytvorení syntetických odtlačkov nasledovala kontrola kvality. Kritérium kvality vygenerovaného odtlačku je rozoznateľnosť od reálneho odtlačku. Pre meranie tohto kritéria bolo vytvorených 6 klasifikátorov. Najlepší klasifikátor dokázal rozoznať reálne odtlačky iba o 0,43 % lepšie ako vytvorené syntetické odtlačky. Vygenerované odtlačky popisuje obrázok 5.3. [28]



Obrázok 5.3: Prvý rad obsahuje reálne odtlačky, druhý rad obsahuje odtlačky vygenerované pomocou SYNFI. Prevzaté a upravené z [28]

## 5.4 SyFDaS

SyFDaS je program poskytujúci generovanie syntetických odtlačkov a simuláciu poškodenia vplyvom ochorenia alebo nekvalitného snímania odtlačku. Generovanie odtlačku začína nastavením vstupných markantov. Ďalej je možné nastaviť rôzne filtre, orientáciu, masku, hustotu línií a iné. Poskytuje taktiež možnosť načítať vlastný odtlačok a do neho potom vkladať poškodenia. Po nastavení parametrov pre vygenerovanie odtlačku nasleduje nastavenie parametrov pre simuláciu poškodenia, ktoré zahŕňa technológiu senzoru, typ snímania a druh poškodenia. Jednotlivé typy senzoru môžu spôsobovať pre nich typické poškodenia. V prípade dotykového senzoru sú to napríklad poškodený senzor, tlak a vlhkosť, skreslenie kože. Medzi ochorenia, ktoré je možné simulovať patria bradavice, atopický ekzém a psoriáza. Obrázok 5.4 ukazuje výstupy simulácie ochorení. [17]

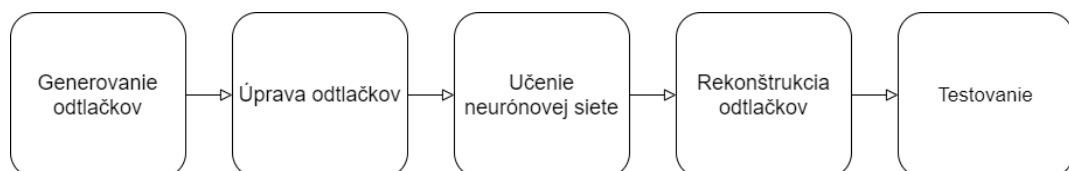


Obrázok 5.4: Simulácia ochorenia pokožky pomocou nástroja SyFDaS.

## Kapitola 6

# Návrh

Proces rekonštrukcie odtlačkov sa skladá z niekoľkých úkonov, ktoré sú zobrazené na obrázku 6.1, prvým je generovanie odtlačkov. Syntetické odtlačky budú generované pomocou nástroja Anguli, nakoľko je voľne dostupný a funkcie, ktoré poskytuje sú dostatočné. Po vygenerovaní odtlačkov prstov nasleduje úprava odtlačkov prstov. Úprava odtlačkov bude zahŕňať zmenu rozlíšenia a vloženie niekoľkých syntetických bradavíc, prípadne iného ochorenia do odtlačku. Bradavice budú získané manuálnym vystrihnutím konkrétneho poškodeného miesta z obrázkov vytvorených programom SyFDaS [18]. Následne budú bradavice vkladané na náhodné miesta v rôznych počtoch. Tento postup vytvorenia vlastného programu, ktorý vkladá poškodenia do odtlačkov prstov bol použitý za účelom zrýchlenia tvorby databáze. Po vytvorení databáze je ďalším krokom učenie neurónovej siete. Po výbere architektúry a implementovaní siete je potrebné sieť naučiť rekonštruovať poškodenia. Sieť dostáva na vstupe dve sady odtlačkov, z ktorých prvá sada sú syntetické odtlačky vytvorené programom Anguli a druhá sada budú tieto odtlačky s vloženými poškodeniami. Výstupom učenia neurónovej siete je súbor s váhami neurónovej siete, ktorý bude použitý pre rekonštrukciu odtlačkov. Rekonštruované odtlačky budú potom testované za účelom analýzy kvality rekonštrukcie.



Obrázok 6.1: Návrh procesu rekonštrukcie syntetických odtlačkov prstov.

### 6.1 Získanie a úprava odtlačkov

Ak chceme učiť sieť na rekonštrukciu poškodení, potrebujeme získať čo najväčšiu databázu odtlačkov. Veľkosť databáze je kľúčová a patrí medzi základ kvalitne naučenej siete. Získanie dostatočnej databáze odtlačkov môže byť problém. Voľne dostupné databáze odtlačkov sú veľmi malé a nejednotné. Taktiež môže nastať situácia kedy by sa vytvorená neurónová sieť nedokázala naučiť rekonštruovať reálne odtlačky vzhľadom na ich veľkú variabilitu a komplexnosť. Z toho dôvodu bude prvá fáza učenia siete prebiehať na syntetických odtlačkoch, ktoré sú dostupné v neobmedzenom množstve a ich jednoduchosť zjednoduší generalizáciu siete. Po úspešnom naučení siete rekonštruovať syntetické odtlačky bude druhá fáza učenia



prebiehať na reálnych odtlačkoch. Do odtlačkov budú vkladané poškodenia za účelom simulácie ochorenia pokožky. Táto simulácia ochorenia umožní vkladat do odtlačku poškodenia rôznych veľkostí, na začiatok je vhodné naučiť sieť rekonštruovať iba malé časti a zistiť tak či sieť dokáže vôbec zvýšiť kvalitu odtlačku prstu. Následne bude možné vkladat väčšie ochorenia a poškodenia. Získané odtlačky môžu disponovať rôznymi rozlíšeniami a môžu byť uložené v rôznych formátoch. Prvotná úprava obrázka s odtlačkom bude nevyhnutná.

### 6.1.1 Syntetické odtlačky

Generovanie syntetických odtlačkov poskytuje neobmedzene veľkú databázu. Stanovenie veľkosti potrebného počtu vzoriek bude prebiehať sledovaním výstupov siete pre rôzne veľkosti. Ak nebude zaznamenaná zmena pri zväčšení datasetu, vieme že nie je potrebné už pridávať nové vzorky. Syntetické odtlačky sú oproti reálnym odtlačkom veľmi konzistentné, preto databáza takýchto odtlačkov nebude musieť byť taká veľká ako v prípade reálnych odtlačkov. Voľne dostupným programom Anguli je možné jednoducho generovať odtlačky. Výstupom tohto programu sú šablóny pre jednotlivé odtlačky a aj variácie odtlačkov z nich vytvorených, znázornené na obrázku 6.2. Šablóny sú konzistentnejšie vo farbe a tvare, a pre neurónovú sieť bude jednoduchšie rekonštruovať tieto šablóny. Za účelom postupného vylepšovania siete bude najprv naučená na týchto šablónach.



Obrázok 6.2: Ukážka šablóny a z nej vytvoreného odtlačku, pomocou programu Anguli.

### 6.1.2 FVC2002

Táto databáza obsahuje odtlačky získané optickým senzorom. Databáza bola vytvorená nasnímaním 110 prstov, z čoho každý bol nasnímaný 8 krát z rôznych uhlov a pod rôznym tlakom, celkovo teda 880 odtlačkov. Databáza obsahuje veľmi nekonzistentné vzorky. Odtlačok častokrát zaberá iba malú plochu obrázka a nie je kompletný. Pre spracovanie tejto databázy bude potrebné odtlačky vystrihnúť a upraviť ich rozlíšenie.

## 6.2 Výber architektúry

Výber správnej architektúry neurónovej siete je dôležitý. Každá architektúra disponuje jedinečnými vlastnosťami, ktoré treba zohľadniť pri jej výbere. Naskytajú sa dve architektúry, ktoré môžu byť použité na rekonštrukciu odtlačkov autoencoder a GAN. V nasledujúcej podkapitole budú vysvetlené princípy ich fungovania a používania.



Obrázok 6.3: Niektoré odtlačky z databáze FVC2002.

### 6.2.1 Autoencoder

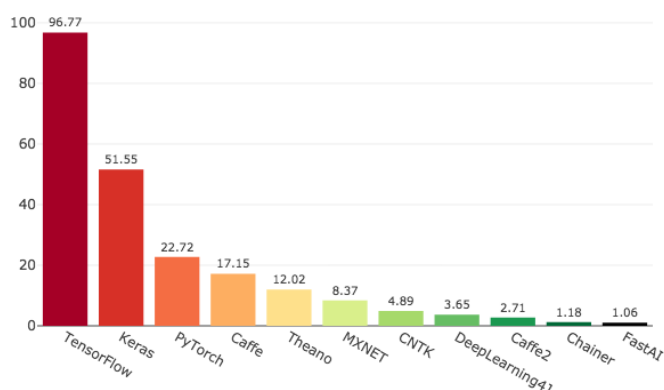
Tento typ architektúry sa skladá z dvoch častí encoder a decoder. Decoder má za úlohu získať vzory z obrázku a zmenšovať obrázok. Decoder potom zoberie výsledky z encoderu a pokúsi sa rekonštruovať vstup. Encoder si môžeme predstaviť ako konvolučnú sieť s vrstvami tak ako sú popísane vyššie. Decoder je potom inverzná verzia encoderu. To znamená že obsahuje také isté vrstvy ale namiesto downsamplingu sa tu nachádza upsampling. Týmto spôsobom môžeme z multidimenziálneho vektora vzorov, ktorý získal encoder, vytvoriť pôvodný obrázok. Ďalším typickým príkladom pre použitie takejto architektúry je sémantická segmentácia. V tomto prípade sieť každému pixelu priradí objekt. Je to jeden z dôležitých problémov počítačového videnia [21].

### 6.2.2 GAN

GAN (Generative adversarial network, vo voľnom preklade generatívna konfrontačná sieť), je typ architektúry, ktorý podobne ako autoencoder má dve časti. Prvou je **generátor**, táto časť má za úlohu previesť vstup, zvyčajne v podobe šumu, na žiadaný výstup. K tomu mu pomáha druhá časť siete - **diskriminátor**, jeho úlohou je vyhodnocovať výstupy generátoru. Porovnáva výstupy generátoru s tréningovými dátami, tak že označí jeho výstup za pravdivý alebo falošný. Úlohou generátoru je zväčšovať chybu diskriminátoru, tým že bude generovať čo raz viac podobné výstupy podobajúce sa tréningovej sade. Takýto typ siete je teda veľmi vhodný pre generovanie nových výstupov. Jeden z možných použití je generovanie odtlačkov. Práca [30] popisuje generovanie nových odtlačkov prstov, podobné tým z tréningovej množiny. [10]

## 6.3 Výber knižnice

Po výbere architektúry, je potrebné zvážiť možnosti knižníc, ktoré budú použité pre vytvorenie siete. Na výber je niekoľko voľne dostupných knižníc. Každá knižnica poskytuje iné prostredie pre vývoj aplikácií. Pre zmeranie výkonu knižnice je potrebné sa zamerať na niekoľko vecí napríklad veľkosť užívateľskej základne či počet odborných článkov, ktoré spomínajú danú knižnicu. Medzi najvýkonnejšie knižnice patria TensorFlow a Keras. Výkon niektorých knižníc ukazuje obrázok 6.4.



Obrázok 6.4: Skóre výkonu jednotlivých knižníc, upravené a prevzaté z [12].

### 6.3.1 TensorFlow

TensorFlow, bol vyvinutý tímom Google Brain<sup>1</sup>. Je to voľne dostupná knižnica vytvorená pre numerické výpočty a strojové učenie veľkého rozsahu. Okrem iného poskytuje zdroje pre vytváranie hlbokých neurónových sietí. TensorFlow dokáže trénovať neurónové siete s rôznym využitím. Práca s touto knižnicou spočíva vo vytváraní diagramu dátového toku, ktorý popisuje pohyb týchto dát cez jednotlivé uzly. Každý uzol reprezentuje súbor matematických operácií a každé prepojenie hrán je multidimenzionálne pole taktiež nazývané tenzor. Výhodou tejto knižnice je veľká miera abstrakcie s akou je možné vyvíjať aplikácie strojového učenia. Používa jazyk python ako jeho rozhranie, avšak zložité matematické výpočty sú implementované v jazyku C++. Aplikácie vyvinuté pomocou tejto knižnice je možné spustiť na rôznych platformách, medzi ktoré patria aj cloudy<sup>2</sup> či iOS a Android zariadenia. Google ponúka svoj vlastný cloud vybavený hardvérom, ktorý bol optimalizovaný pre vývoj tensorflow aplikácií. Tensorflow ponúka taktiež interaktívne grafické rozhranie, v ktorom je možné vyhodnocovať a modifikovať jednotlivé uzly grafu. Jednou z nevýhod tejto knižnice je jej nedeterministickosť. Kvôli jej implementácii sa môže stať že učenie rovnakých modelov na rozdielnych systémoch bude vykazovať odlišné výsledky. Tento problém vychádza hlavne z odlišností jednotlivých druhov grafických kariet [24]. [8]

### 6.3.2 Keras

Táto knižnica sa zakladá na jednoduchosti používania, tak aby strojové učenie priblížila čo najviac ľuďom. Ponúka konzistentné a jednoduché rozhranie tak aby minimalizoval nevyhnutné úkony pre vytváranie neurónových sietí. Podrobné a jasné chybové hlášky zjednodušujú ladenie programov. Keras je integrovaný v rámci knižnice TensorFlow, viď podkapitola 6.3, toto umožňuje implementáciu zložitejších štruktúr v rámci používania jednoduchého rozhrania. Ako aj TensorFlow, aj túto knižnicu je možné používať na viacerých platformách, medzi ktoré patria napríklad: iOS, Android, Google Cloud, Raspberry Pi<sup>3</sup>. Modely vytvorené použitím tejto knižnice je možné trénovať na rôznom hardvéri, okrem procesorov sú

<sup>1</sup><https://research.google/teams/brain/>

<sup>2</sup><https://cloud.google.com/>

<sup>3</sup><https://www.raspberrypi.org/>

to grafické karty od NVIDIA ale aj AMD [33]. Podporuje aj paralelné učenie na viacerých grafických kartách naraz <sup>4</sup>.

## 6.4 Úprava siete, učenie a rekonštrukcia

Po výbere architektúry je nutné sieť naučiť vykonávať rekonštrukciu čo najkvalitnejšie. Pri učení siete je máme niekoľko možností ako sledovať a upravovať sieť tak aby poskytovala čo najlepšie výsledky. Ideálne naučený model je ten, ktorý je na rozmedzí *underfittingu* a *overfittingu*. Aby sme zistili kde končia možnosti siete, musíme sieť dostať za hranicu *overfittingu*. Medzi techniky, ktorými toho možno dosiahnuť patria:

- Pridávanie a odoberanie vrstiev,
- zväčšovanie vrstiev,
- zmena počtu epoch.

Po každej zmene nejakého z týchto parametrov je nutné sledovať *loss* funkciu pre tréningové a validačné dáta. Ak začneme pozorovať zvyšovanie *loss* funkcie validačných dát, vieme že sme dosiahli *overfitting* a od tohto bodu bude sieť podávať horšie výsledky pri ďalšom učení. Ďalšou fázou úpravy siete bude zmena jej parametrov tak aby sme dostali čo najlepší model. [3]

---

<sup>4</sup>[https://keras.io/why\\_keras/](https://keras.io/why_keras/)

# Kapitola 7

## Implementácia

Ako prvé je potrebné vybrať typ neurónovej siete, ktorá sa bude učiť odtlačky rekonštruovať. Na výber je niekoľko druhov knižníc a architektúr sietí, ktoré je možné použiť. V predchádzajúcej kapitole boli popísané niektoré typy architektúr, ktoré môžu byť použité. Architektúra GAN je veľmi vhodná pre generovanie nových výstupov, ktoré sa nenachádzajú medzi trénovacou množinou. Z tohoto dôvodu bude použitá v prvej verzii aplikácie architektúra autoencoder. Pre jednoduché používanie tejto aplikácie bude vytvorené grafické užívateľské rozhranie v jazyku Python. Veľmi dôležitou súčasťou je príprava trénovacej sady, ktorá bude popísaná v tejto kapitole.

### 7.1 Vytváranie databáze odtlačkov

Celkovo bolo vytvorených 5 databáz. Syntetické odtlačky boli generované pomocou programu Anguli. Reálne odtlačky pochádzajú z databáze FVC2002. Do každého odtlačku bolo vložených niekoľko bradavíc. Taktiež bola potrebná úprava rozlíšenia a zmena formátu, v ktorom sú odtlačky uložené.

Do odtlačkov boli vkladané syntetické bradavice. Bolo vytvorených 10 syntetických bradavíc z týchto bradavíc bolo vytvorených ďalších 30 bradavíc, pričom každá bola rotovaná o 90°. Celkovo bolo vytvorených 40 bradavíc o priemere približne 80 pixelov. Z tejto sady pri vkladaní vyberie náhodná bradavica a vloží sa na náhodné miesto odtlačku.

#### 7.1.1 MasterFingerprints

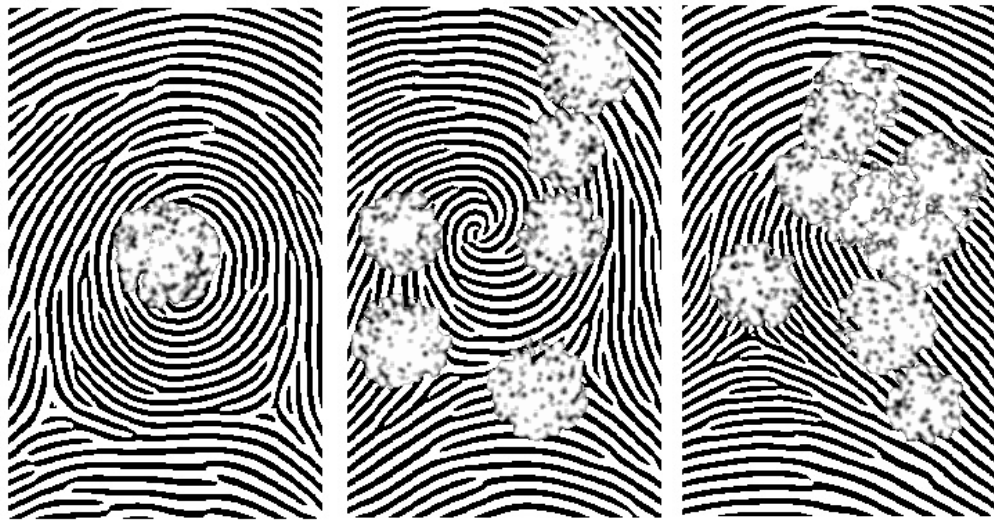
Výstupom programu Anguli sú dve sady syntetických odtlačkov, prvá obsahuje takzvané *master fingerprints*, tieto odtlačky sú akousi šablónou pre vytváranie syntetických odtlačkov. Tento *master fingerprint* neobsahuje žiadny šum, transláciu či rotáciu a má tvar obdĺžnika. Pre neurónovú sieť tak bude jednoduchšie rekonštruovať tieto odtlačky preto je vhodná pre počiatočné fáze učenia a ladenia siete.

Vygenerované odtlačky majú rozlíšenie  $400 \times 275$  pixelov, tento rozmer nie je vhodný *downsampling* vrstvy, keďže každá táto vrstva mení rozmery vstupu tak, že na výstup má dva-krát menšie rozmery. Rozmer 275 pixelov sa nedá rozdeliť na dve časti bez straty informácie. Urobiť *downsampling* pri takomto rozmere je možné, avšak na výstupe by sme dostali rozmer 137 pixelov, ak by sme chceli naopak robiť *upsampling* z tohoto rozmeru, dostali by sme hodnotu 274 pixelov čo nie je pôvodná hodnota rozlíšenia. Z tohoto dôvodu musí byť rozlíšenie upravené. Python knižnica PILLOW, ponúka množstvo funkcií pre spracovanie obrázkov. Pomocou funkcie `resize()` zmeníme rozlíšenie odtlačku na hodnotu  $400 \times 256$

pixelov, Nové rozlíšenie obrázka  $400 \times 256$  je vhodné pre konvolučnú neurónovú sieť, pri tomto rozlíšení je možné robiť *downsampling* a *upsampling* až 4-krát bez straty informácie.

Formát vygenerovaného obrázku v prípade programu Anguli je voliteľný, môžeme vybrať z niekoľkých formátov avšak všetky ponúkané formáty obsahujú farebné kanály, ktoré nie sú potrebné keďže neurónová sieť pracuje s obrázkami v odtieňoch šedej farby. Ďalej formát môže obsahovať aj takzvaný *alpha* kanál, ktorý určuje priehľadnosť pixelov. Všetky nepotrebné farebné kanály budú teda odstránené. Algoritmus, ktorý odstraňuje tieto kanály prechádza každý pixel obrázku a uloží iba odtieň šedej, zostávajúce farebné kanály nie sú použité. Po úprave rozlíšenia nasleduje vkladanie bradavíc na náhodné miesta.

Celkovo boli vytvorené tri sady, prvá sada obsahuje jednu vloženú bradavicu, druhá sada obsahuje tri a tretia sada obsahuje desať vložených bradavíc. Každá sada obsahuje 2000 odtlačkov, z toho polovica bude použitá na trénovanie a validáciu, a zvyšných 1000 odtlačkov bude použitých na testovanie. Vytvorené sady ilustruje obrázok 7.1.



Obrázok 7.1: Ukážka syntetických *master fingerprint* odtlačkov s vloženými bradavicami.

### 7.1.2 Impressions

Ďalšou vytvorenou sadou sú syntetické odtlačky vytvorené odvodením z *master fingerprintov* a nazývajú sa *impressions*. Tieto odtlačky už obsahujú určitý šum, nachádzajú sa v nich aj náznaky potných pórov. Niektoré odtlačky sú aj mierne rotované. Tak isto ako u predchádzajúcej sade, bolo treba zmeniť rozlíšenie pomocou `resize()`, zmeniť formát obrázku na 8-bitový čiernobiely. Do každého odtlačku boli vložené 4 bradavice, pričom miesto polohy bradavice bolo oproti predchádzajúcej sade, posunuté viac do stredu tak aby bradavica nepresahovala cez hranice nepravidelného tvaru odtlačku. Bolo vytvorených 2000 odtlačkov tejto sady, prvá polovica bude použitá na trénovanie a validáciu, a zvyšných 1000 bude použitých na testovanie. Niektoré vytvorené odtlačky sú znázornené na obrázku 7.2.

### 7.1.3 FVC

Ďalšou sadou obrázkov sú reálne odtlačky z databázy FVC2002db1. Úpravy tejto databázy boli takmer rovnaké ako u predchádzajúcich. Bola potrebná úprava formátu na čiernobiely,



Obrázok 7.2: Ukážka syntetických odtlačkov s vloženými bradavicami.

zmena rozlíšenia a vloženie bradavice. Pôvodné rozlíšenie  $388 \times 372$  pixelov bolo zmenené na  $400 \times 400$  pixelov pomocou už spomínanej funkcie `resize()`. Jedným z argumentov tejto funkcie je určenie prevzorkovania nových pixelov, implicitná hodnota je použitie farby najbližšieho pixelu, pri tejto variante sady bol navyše po prevzorkovaní použitý vyhladzovací filter `Image.ANTIALIAS`, ktorý obmedzil vznik artefaktov. Bradavica bola vložená na náhodné miesto obrázka. Odtlačky ale majú rôznu polohu v rámci obrázku a náhodná pozícia bradavice bola v niektorých prípadoch určená mimo odtlačku. Takto chybné vytvorené odtlačky boli vyradené a vkladanie bradavice sa opakovalo až kým nedošlo k správne umiestneniu na plochu odtlačku. Veľkosť tejto databázy 880 odtlačkov nemusí však byť dostatočná pre kvalitné naučenie neurónovej siete. Odtlačky tejto databázy sú veľmi variabilné, niektoré odtlačky boli snímané pod príliš veľkým tlakom. Niektoré obrázky z vytvorenej sady popisuje obrázok 7.3.



Obrázok 7.3: Ukážka reálnych odtlačkov z databázy fvc2002db1 s vloženými bradavicami.

## 7.2 Učenie siete

Učenie neurónovej siete požaduje pomerne vysoký výpočetný výkon. Táto sieť sa učila na zostave s grafickou kartou NVIDIA GTX1050 ti so štyrmi gigabajtami vlastnej pamäte, procesorom Intel i5-2400 a ôsmimi gigabajtami pamäte RAM. Jedna, použitie tejto grafickej trvá približne 13 sekúnd, pre 300 epoch je doba učenia zhruba hodina.

Prvým krokom učenia siete je načítanie odtlačkov z vytvorenej sady do podoby vhodnej pre sieť. Obrázok je načítaný jeden po druhom a zároveň je aj spracovaný. Obrázok je konvertovaný do formátu poľa pomocou funkcie `numpy.asarray()`. Toto pole obsahuje hodnoty v rozmedzí  $<0,255>$ . Neurónová sieť pracuje s hodnotami v intervale  $<0,1>$ , obrázok je teda prevedený na dátový typ `float32` a jednotlivé hodnoty v poli sú vydelené maximálnou hodnotou v poli. Dostávame tak normalizované hodnoty  $<0,1>$ . Pomocou funkcie `reshape()` vytvoríme 4-rozmerné pole výsledných obrázkov. Prvý rozmer je index jednotlivých obrázkov, druhý a tretí rozmer je šírka a výška, štvrtý rozmer predstavuje hodnoty pixelov. Takto upravené obrázky sú pripravené ako vstup do siete. Celá funkcia pre načítanie obrázkov je znázornená v zdrojovom kóde 7.1.

```
def loadImages(path):
    images = []

    for file in os.listdir(path):
        image = Image.open(path+file)
        image = numpy.asarray(image)
        image = image.astype('float32')
        images.append(image)

    images = np.array(images)
    images = images.reshape(-1, 400, 256 ,1)
    images = images / numpy.max(images)
    return images
```

Zdrojový kód 7.1: Funkcia pre načítanie a spracovanie obrázkov z datasetu.

Po spracovaní obrázkov nasleduje rozdelenie datasetu na dve časti, tréningovú a validačnú, pomocou funkcie `train_test_split()`. Táto funkcia prijíma niekoľko parametrov. Povinnými parametrami sú polia, ktoré chceme rozdeľovať. Prvým je v tomto prípade sada odtlačkov s vloženými bradavicami, druhá je sada odtlačkov bez poškodenia. Ďalším parametrom je pomer rozdelenia celej sady na validačnú a tréningovú, v tomto prípade bola použitá hodnota 0,2, čo znamená že 20 % odtlačkov bude patriť do validačnej sady.

Ako východzia architektúra bola použitá sieť autoencoder z článku [2], znázornená v zdrojovom kóde 7.2, pôvodný počet vrstiev a filtrov bol zachovaný. Prvé učenie prebiehalo pomocou tejto siete. Ako optimalizátor bol použitý `RMSprop()`. Veľkosť maximálnej dávky je určená veľkosťou pamäte grafickej karty, v tomto prípade to je maximálna dávka 21 vzoriek.

```
x = Conv2D(16, (3, 3), activation='relu', padding='same')(input)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
```



```

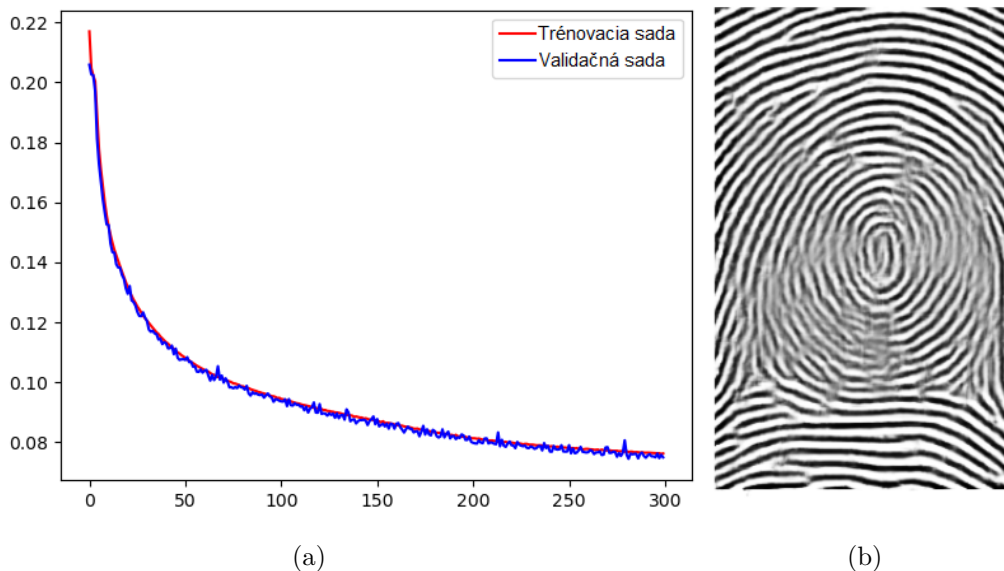
encoded = MaxPooling2D((2, 2), padding='same')(x)

x = Conv2D(8, (3, 3), activation='relu', padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(16, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

```

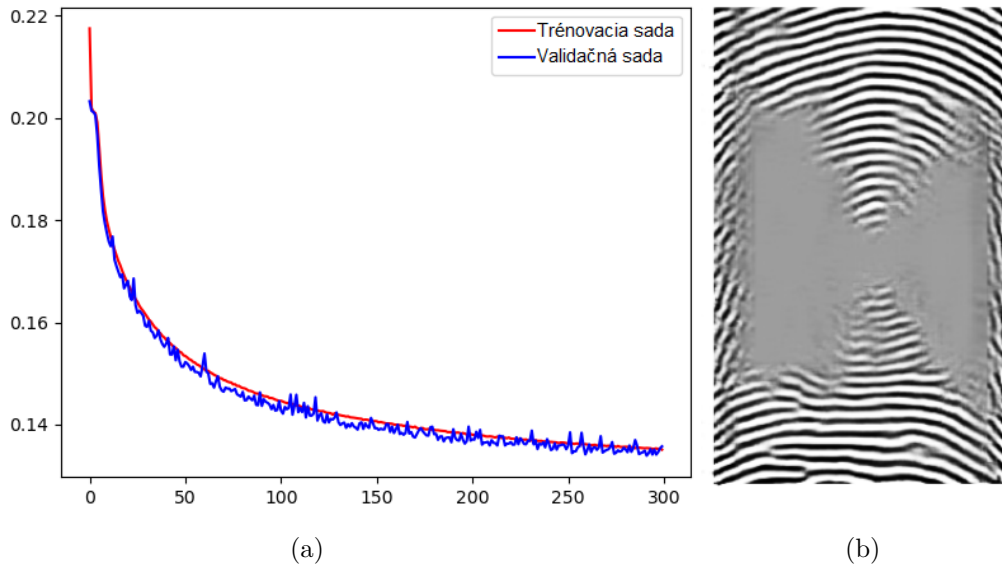
Zdrojový kód 7.2: Pôvodná sieť autoencoderu s 8 vrstvami.

Cieľom prvého učenia je dosiahnuť *overfitting* veľkým počtom epoch, bola teda zvolená hodnota 300 epoch. Výsledky *loss* funkcie a výstupu siete vidíme na obrázku 7.4. Z tohoto učenia je potrebné si všimnúť že *loss* funkcia validačnej aj trénovacej sady stále klesá, *overfitting* by teda nemusel vôbec nastať ani po ďalších epochách. Je to z dôvodu že sieť sa naučila rekonštruovať obrázky tak aby sa chyba znižovala. V tomto prípade sa síce chyba znižuje ale na úkor správnosti rekonštrukcie. Markanty nie sú zachovávané a línie sú rozmazané. Tato sieť doplní šedý pixel na určité miesto a tým zníži výslednú *loss* funkciu, je teda neschopná akejkoľvek rekonštrukcie. Ďalšia úprava siete sa bude týkať počtu vrstiev.



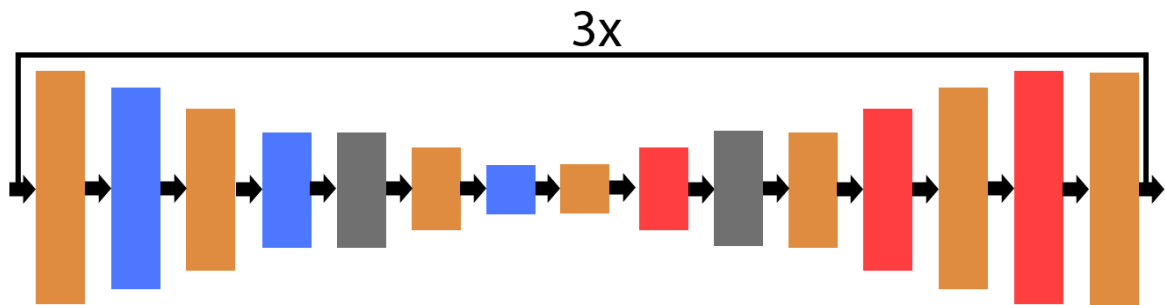
Obrázok 7.4: Loss funkcia prvej použitej neurónovej siete (a), výstup rekonštrukcie odtlačku, pomocou tejto siete (b).

Po neúspešnej prvej architektúre siete je potrebné vykonať úpravu. Medzi základné možné úpravy, ako už bolo spomínané, patria zmeny počtu vrstiev alebo filtrov. Do siete boli pridané štyri vrstvy, dve vrstvy do encoderu a dve do decoderu. Do encoderu bola pridaná konvolučná vrstva a downsampling vrstva, do decoderu bola pridaná konvolučná vrstva a upsampling vrstva. Ani v prípade takejto architektúry nedošlo k *overfittingu* viď obrázok 7.5a. Sieť sa naučila minimalizovať chybu opäť vkladaním šedých pixelov ako znázorňuje obrázok 7.5b. Ďalšou zmenou v architektúre bude zväčšovanie počtu filtrov v konvolučných vrstvách.



Obrázok 7.5: Loss funkcia druhej použitej neurónovej siete (a), výstup rekonštrukcie od-  
tlačku, pomocou tejto siete (b).

Predchádzajúce architektúry zlyhávali v správnej rekonštrukcii, pridávanie vrstiev ne-  
zlepšovalo výstupy siete. V ďalších testoch museli byť navýšené počty filtrov, keďže pôvodná  
sieť pracovala s obrázkami s veľmi nízkymi rozmermi -  $28 \times 28$  pixelov, oproti syntetickým  
odtlačkom s rozmermi -  $400 \times 256$  pixelov. Po niekoľkých desiatkach pokusov bola navýšené  
počty filtrov v konvolučných vrstvách na hodnoty: 32, 64, 128 v decoderi a 128, 64, 32, 1 v  
encoderi. Nasledujúci obrázok 7.6 popisuje konečnú vytvorenú architektúru. Sieť kvalitnejšie  
spracováva vstupy a nevytvára taký veľký šum ako v predchádzajúcich prípadoch.



Obrázok 7.6: Výsledná architektúra neurónovej siete. Modré vrstvy predstavujú *downsam-  
pling*, červené *upsampling*, oranžové konvolučné vrstvy a šedé dropout vrstvy.

Učenie je treba zastavovať v epoche, v ktorej hodnota *loss* funkcie pre validačnú sadu  
dosahuje globálne minimum. Zachovanie váh tejto epochy je možné realizovať funkciou  
`ModelCheckpoint()`. Pomocou tejto funkcie môžeme ukladať váhy modelu po každej epoche  
alebo nastaviť ukladanie váh iba po zlepšení modelu, čiže po znížení *loss* hodnoty pre  
validačnú sadu. Ak nastavíme ukladanie váh iba po znížení tejto hodnoty, na konci učenia  
dostávame váhy, ktoré sú na hranici *overfittingu* a *underfittingu*.

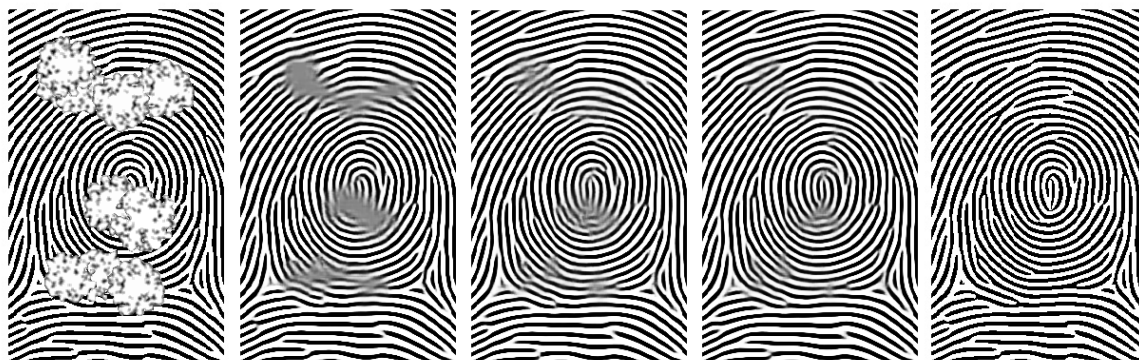
Výsledky výstupu siete sú však často rozmazané a obsahujú množstvo nových chýb, ktoré tam boli vložené v procese rekonštrukcie. Ani táto architektúra teda nestačila na kvalitnú rekonštrukciu. Poslednou zmenou bola iteratívna rekonštrukcia odtlačku. Sieť bola učená v troch iteráciách. Týmto spôsobom simulujeme sieť, ktorá je  $3\times$  väčšia. Dôvod takejto simulácie je obmedzená veľkosť pamäte grafickej karty (4GB). Ak by sme použili sieť takejto veľkosti, znížila by sa maximálna veľkosť dávky na 7 vzoriek. Pri takejto malej veľkosti dávky sa sieť nie je schopná naučiť kvalitne rekonštruovať.

Tento postup troch iterácií bol zopakovaný aj v prípade učenia siete na ostatných vytvorených sadách s tromi a šiestimi bradavicami, a taktiež na dátovej sade *fv2002* a *impressions* sade. Dôležitou súčasťou novej architektúry siete sú *dropout* vrstvy. Ich použitím sa predĺžila fáza učenia kedy nastával *underfitting*. Sieť sa tak dokázala učiť o niekoľko desiatok epoch viac až na hranicu *overfittingu*. Taktiež sa znížili hodnoty *loss* funkcií pre validačné sady. Nasledujúca tabuľka 7.1 ukazuje počty epoch, počas ktorých sa sieť učila. Ďalšie učenie už zvyšovalo hodnoty *loss* funkcií pre validačné sady.

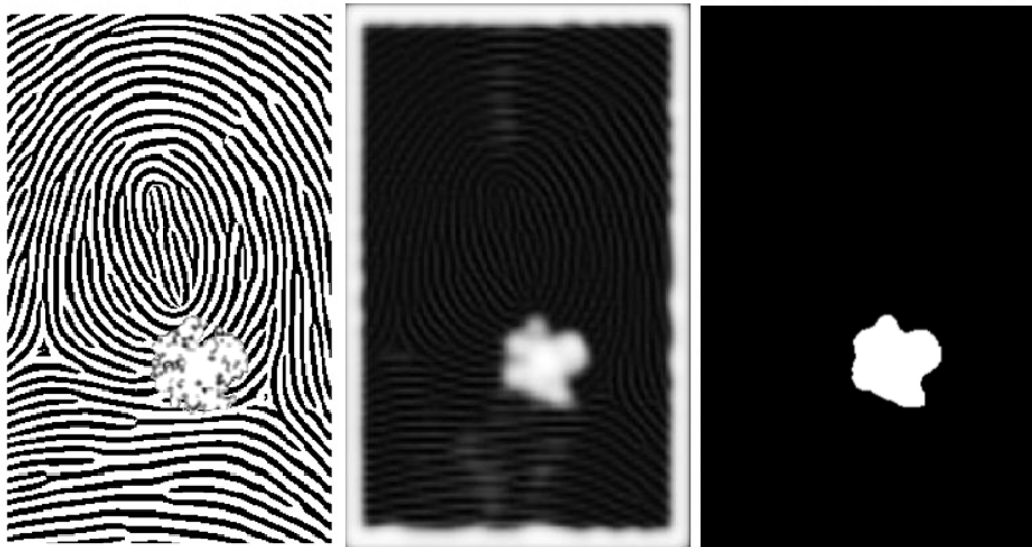
Tabuľka 7.1: Počty epoch, po ktorých bolo učenie zastavené pre jednotlivé sady.

sada	1. iterácia	2. iterácia	3. iterácia
master fingerprint +1 bradavica	298	108	118
master fingerprint +6 bradavíc	241	92	101
master fingerprint +10 bradavíc	285	222	121
impression fingerprint +4 bradavice	199	111	103
fv2002 +1 bradavica	258	173	135

Proces rekonštrukcie pomocou troch iterácií je dobre zobrazený na obrázku 7.7. Po prvej iterácii rekonštrukcie vidíme že sieť zafarbí bradavicu na šedo a zachová vstupy línií do bradavice a výstupy línií z bradavice. V druhej iterácii sa z týchto šedých miest vytvoria línie. V tretej iterácii sa vyhladia línie a zviditeľnia sa ryhy.



Obrázok 7.7: Proces rekonštrukcie odtlačku v jednotlivých iteráciách dátovej sady s desiatimi bradavicami. Prvý odtlačok je poškodený bradavicami, nasledujúce tri obrázky sú výstupy siete po prvej, druhej a tretej iterácii. Piaty obrázok je originálny odtlačok bez poškodenia.



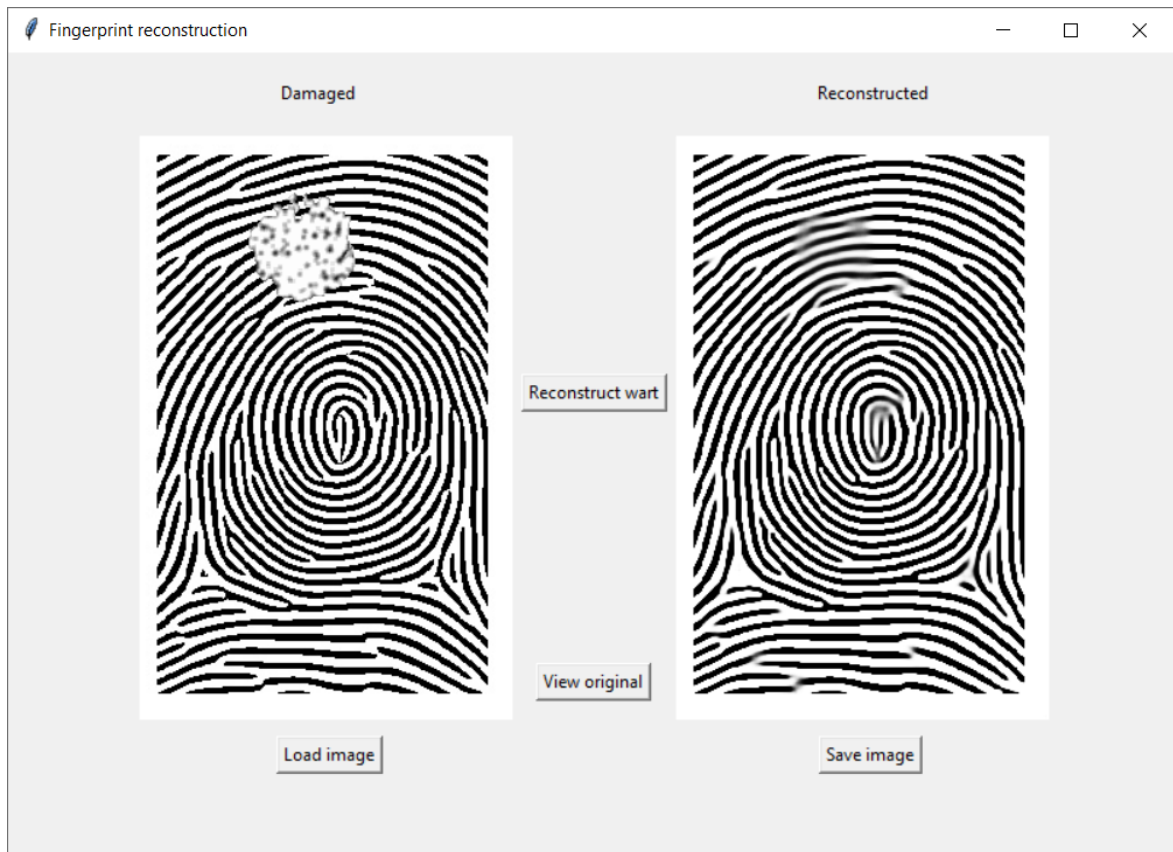
Obrázok 7.8: Ukážka detekcie bradavice. Zľava do prava: poškodený odtlačok, predikcia rekonštrukcie a úprava predikcie filtrom.

### 7.2.1 Detekcia poškodenia

Pri experimentovaní s učením siete vznikol jednoduchý detektor polohy a veľkosti bradavice. Tento detektor používa rovnakú sieť ako pre rekonštrukciu, avšak je naučený iba jednou epochou. Takáto sieť ešte nevie čo sa od nej očakáva preto oblasť s bradavicou celú zafarbí na šedo a okolie na čierne. Pomocou filtra teraz môžeme bradavicu zvýrazniť a zistiť tak jej polohu a tvar. Ukážka postupu detekcie bradavice je znázornená na obrázku 7.8. Detektor slúži predovšetkým ako ilustrácia možností neurónových sietí a je jedným z možných rozšírení do budúcnosti.

## 7.3 Grafické užívateľské rozhranie

Pre demonštráciu aplikácie bolo vytvorené jednoduché užívateľské rozhranie. Hlavné okno obsahuje dve menšie okná, v ktorom sa zobrazuje obrázok, ktorý bude rekonštruovaný a vedľa toho obrázok, ktorý bol zrekonštruovaný pomocou natrénovanej neurónovej siete. Užívateľ si načíta odtlačok do prvého okna. Následne môže rekonštruovať obrázok. V druhom okne sa mu zobrazí výsledok rekonštrukcie, pokiaľ užívateľ nie je s výsledkom spokojný, môže rekonštruovať obrázok opätovne až do tretej iterácie. V tomto prípade zoberie program už rekonštruovaný obrázok a pokúsi sa ho rekonštruovať znova. V prípade že obrázok je jemne zašumený, dokáže program vyhladiť hrany a odstrániť šum. Ak je užívateľ s výsledkom spokojný, môže si obrázok uložiť. Ďalšou funkciou je možnosť prezerať originálny odtlačok, v záujme porovnania markantov originálneho a rekonštruovaného odtlačku. Vytvorené grafické užívateľské rozhranie ukazuje obrázok 7.9.



Obrázok 7.9: Grafické užívateľské rozhranie pre rekonštrukciu odtlačkov.

## Kapitola 8

# Vyhodnotenie úspešnosti rekonštrukcie

V tejto kapitole budú zhrnuté výsledky merania kvality a porovnávacieho hodnotenia vzoriek originál-poškodený a originál-rekonštruovaný. Testy rekonštrukcie prebiehali na testovacej sade, ktorej vzorky neboli použité na učenie siete. Bolo použitých vždy 40 vzoriek z každej sady, na ktorých boli merané hodnoty kvality a porovnávacieho skóre. Pomocou nástrojov NFIQ2, FiQiVi 2.0 a VeriFinger 10.0 boli analyzované výstupy neurónovej siete. Niektoré prípady, v ktorých rekonštrukcia prebehla s pomerne dobrými výstupmi sú zobrazené na obrázku 8.1.

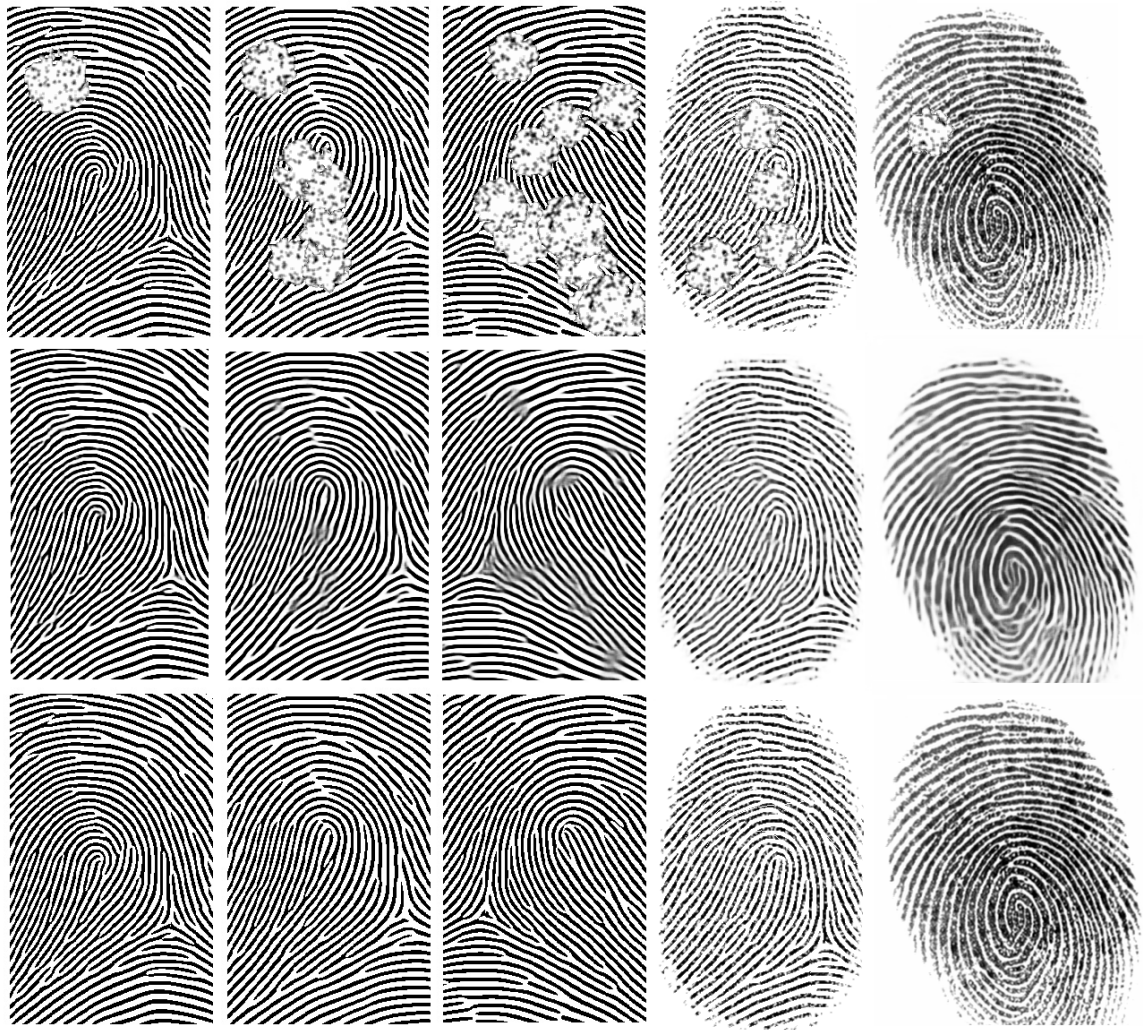
Nie vždy sa rekonštrukcia podarí. Kvalita rekonštrukcie závisí na mieste a veľkosti bradavice. Rekonštrukcia poškodených častí v delte odtlačku je často deformovaná viď obrázok 8.2.

### 8.1 NFIQ2

Prvé testovanie výstupov prebiehalo pomocou nástroja NFIQ2. Tento program prijíma vstupné odtlačky vo formáte BMP (Bitmap image file) alebo WSQ (Wavelet Scalar Quantization). Pri použití formátu BMP však program nameral hodnoty 255 pre poškodený, originálny aj rekonštruovaný odtlačok, preto boli odtlačky prevedené do formátu WSQ. Jedným z argumentov programu NFIQ je mód hodnotenia viacerých (BATCH) odtlačkov alebo iba samostatného SINGLE odtlačku. Pri pokusoch hodnotenia viacerých odtlačkov naraz nastávala chyba `unknown error`. Odtlačky boli hodnotené jeden po druhom v móde SINGLE. Nasledovalo meranie kvality odtlačku. Meranie prvej sady však odhalilo veľkú nekonzistenciu v spracovaní odtlačkov pomocou tohoto nástroja. Tabuľka 8.1 ukazuje výsledky hodnotenia pre sadu s jednou bradavicou. Niekoľko krát sa stalo že poškodený odtlačok bol vyhodnotený ako kvalitnejší oproti originálnemu odtlačku. Priemer hodnoty kvality poškodených odtlačkov je tým pádom vyšší ako priemer hodnoty kvality originálnych odtlačkov. Vyhodnotenie ostatných sád pre jednu a tri iterácie ukazujú grafy B.1 a B.2.

Tabuľka 8.1: Priemerné hodnoty kvality rekonštruovaných odtlačkov prstov, merané pomocou nástroja NFIQ2.

sada	poškodený	rekonštruovaný	rozdiel	originál
master fingerprint +1 bradavica	71,48	67,42	-4,05 %	70,1



Obrázok 8.1: Ukážka rekonštrukcie odtlačkov. Prvý riadok zobrazuje vzorky s vloženými bradavicami, druhý riadok rekonštruované obrázky a tretí riadok originálne odtlačky bez poškodenia.



Obrázok 8.2: Ukážka nekvalitnej rekonštrukcie odtlačkov.

## 8.2 FiQiVi

Ďalší nástroj pre hodnotenie kvality odtlačku je FiQiVi. Oproti predchádzajúcemu nástroju, tento prijíma súbory aj vo formáte PNG (Portable Network Graphics), takže konverzia formátu nebola potrebná. Funkcia hodnotenia viacerých odtlačkov naraz v tomto prípade funguje, a výstupy hodnotení sú uložené v súbore formátu CSV (Comma-separated values). Priemerné hodnoty výstupov po troch iteráciách rekonštrukcie sú znázornené v tabuľke 8.3.

Výsledky merania pre všetky sady boli konzistentné a oproti predchádzajúcemu nástroju, rekonštruovaný odtlačok dosahoval vždy vyššie hodnotenie vzhľadom na poškodený odtlačok ako je možné vidieť na grafoch B.3 a B.4. V tomto prípade sa nestalo že poškodený odtlačok mal vyššiu kvalitu ako odtlačkov bez poškodenia. Zlepšenie kvality rekonštrukcie bolo výrazné pre sady so šiestimi a desiatimi bradavicami. Menšie zlepšenie dosahovali všetky ostatné sady. Avšak zlepšenie kvality samotného odtlačku nie vždy zlepšilo kvalitu rekonštrukcie markantov odtlačku.

Výsledky kvality odtlačkov po jednej iterácii rekonštrukcie znázorňuje tabuľka 8.2. Zlepšenie kvality týchto odtlačkov je menšie oproti rekonštruovaným odtlačkom pomocou troch iterácií. Tento trend nemusí platiť však nemusí platiť aj pre ostatné merania.

Tabuľka 8.2: Priemerné hodnoty kvality poškodených a rekonštruovaných odtlačkov po prvej iterácii, percentuálny rozdiel a hodnoty kvality originálnych odtlačkov, merané pomocou programu FiQiVi.

sada	poškodený	rekonštruovaný	rozdiel	originál
master fingerprint +1 bradavica	88,95	92,45	+3,5 %	93,05
master fingerprint +6 bradavíc	63,7	90,53	+26,83 %	93,03
master fingerprint +10 bradavíc	49,13	88,13	+39 %	93,175
impression fingerprint +4 bradavice	58,28	68,35	+10,07 %	82,075
fvc2002 +1 bradavica	55,7	57,88	+2,18 %	65,75

Tabuľka 8.3: Priemerné hodnoty kvality poškodených a rekonštruovaných odtlačkov po tretej iterácii, percentuálny rozdiel a hodnoty kvality originálnych odtlačkov, merané pomocou programu FiQiVi.

sada	poškodený	rekonštruovaný	rozdiel	originál
master fingerprint +1 bradavica	88,95	92,73	+3,76 %	93,05
master fingerprint +6 bradavíc	63,7	92,5	+28,8 %	93,05
master fingerprint +10 bradavíc	49,13	92,6	+43,48 %	93,05
impression fingerprint +4 bradavice	58,28	68,08	+9,8 %	82,075
fvc2002 +1 bradavica	55,7	59,35	+3,65 %	65,75

## 8.3 VeriFinger

Pomocou nástroja VeriFinger boli analyzované hodnoty porovnávacieho skóre vzoriek originál-poškodený a originál-rekonštruovaný. Výsledky hodnotenia zobrazuje tabuľka 8.4. Porovnávacie skóre rekonštruovaných vzoriek sa zhoršilo vo všetkých prípadoch. Dôvod zhoršenia je zavádzanie chýb do rekonštruovaného obrázka. Znamená to že sa vytvárajú nové markanty, prípade zanikajú markanty na miestach, ktoré neboli poškodené. VeriFinger je veľmi



citlivý aj na malé zmeny, stačí zmena niekoľkých pixelov a uhol markantu je zmenený čo znehodnotí hodnotenie rekonštrukcie.

V prípade že rekonštruujeme odtlačok iba jednou iteráciou, miesto bradavice je zafarbené na šedo. Samotný program VeriFinger obsahuje algoritmy pre zlepšenie kvality odtlačku a tieto šedé miesta často dobre zrekonštruje. Týmto spôsobom je možné zlepšiť porovnávacie skóre rekonštruovaného odtlačku oproti poškodenému. Tabuľka 8.5 ukazuje porovnávacie skóre po jednej iterácii rekonštrukcie. Zlepšenie skóre po rekonštrukcii narástlo o 6,46 % v prípade sady s desiatimi bradavicami. Zároveň však vidíme že kvalita (meraná pomocou FiQiVi) nedosahuje také hodnoty ako pri rekonštrukcii tromi iteráciami v tabuľke 8.3. Ostatné sady nedosahovali zlepšenie porovnávacieho skóre, dôvodom je vysoká citlivosť programu Verifinger. Tieto testy ukazujú že je možné zlepšiť porovnávacie skóre na úkor kvality odtlačku a naopak. Aby sa však rekonštrukcia oplatila z pohľadu rekonštrukcie správnosti markantov, musí byť poškodenie veľmi vysoké. Bradavice majú priemer okolo 80 pixelov, to znamená že ich plocha je zhruba 5026 pixelov. Pri počte 10 bradavíc na jednom odtlačku môžu zaberáť až 50 % celej plochy ak sa neprekrývajú. V skutočnosti je toto číslo kvôli prekryvaniu o niečo menšie. Poškodenie teda musí dosahovať o niečo menej ako 50 % aby sa rekonštrukcia oplatila z pohľadu verifikácie odtlačkov. Graf B.5 zobrazuje hodnoty porovnávacieho skóre po jednej iterácii rekonštrukcie. Najvýraznejšie zhoršenie skóre nastalo v prípade sady FVC2002. Reálne odtlačky oproti syntetickým obsahujú množstvo jemných markantov, ktoré boli pozmenené v priebehu rekonštrukcie. Podobné výsledky vidíme aj na grafe B.4, kde po troch iteráciách rekonštrukcie sa porovnávacie skóre tejto sady ešte viac znížilo, kvôli zmenám týchto jemných markantov.

Tabuľka 8.4: Priemerné hodnoty porovnávacieho skóre rekonštrukcie, merané pomocou programu VeriFinger.

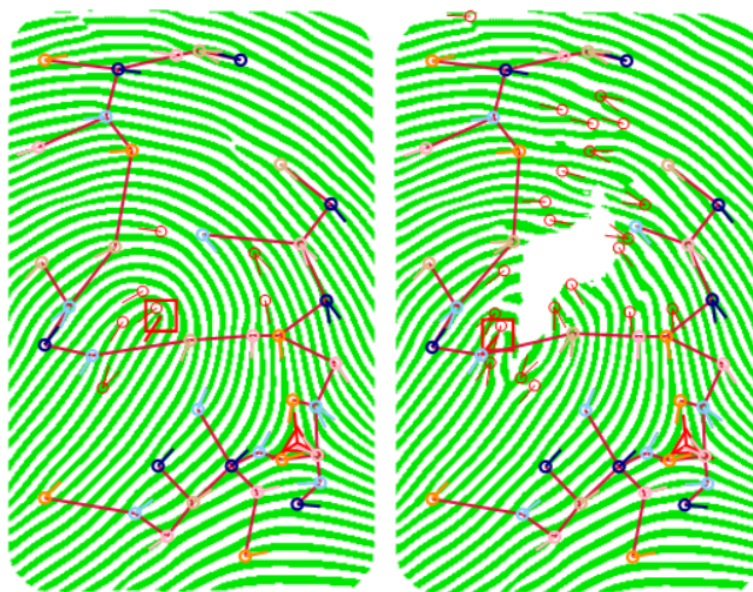
sada	poškodený	rekonštruovaný	rozdiel
master fingerprint +1 bradavica	1291,25	961,775	-25,53 %
master fingerprint +6 bradavíc	756,925	640,175	-15,42 %
master fingerprint +10 bradavíc	559,8	546,075	-2,45 %
impression fingerprint +4 bradavice	848,225	668,15	-21,23 %
fvc2002 +1 bradavica	1383,3	501,65	-63,74 %

Tabuľka 8.5: Priemerné hodnoty porovnávacieho skóre s originálnym odtlačkom, rozdiel hodnotenia a priemerná kvalita rekonštruovaného odtlačku.

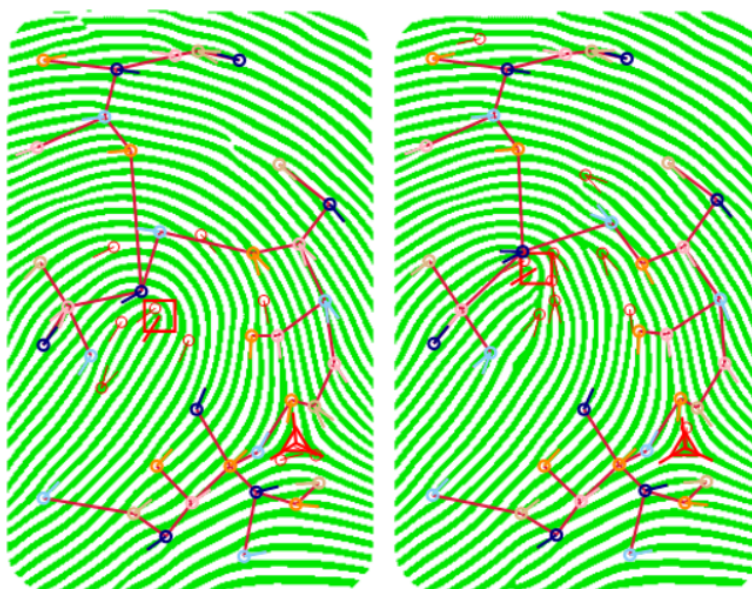
sada	poškodený	rekonštruovaný	rozdiel
master fingerprint +1 bradavica	1291,25	1079,8	-16,38 %
master fingerprint +6 bradavíc	756,93	702,18	-7,23 %
master fingerprint +10 bradavíc	559,8	595,98	+6,46 %
impression fingerprint +4 bradavice	848,23	737,7	-13,03 %
fvc2002 +1 bradavica	1383,3	653,35	-52,77 %

Výsledky testov ukazujú že porovnávacie skóre nenarastá aj napriek obnoveniu línií v poškodenom odtlačku. Tieto výsledky sú spôsobené špecifickým algoritmom, ktorý je veľmi citlivý aj na najmenšie odchýlky v línách. Program VeriFinger pri porovnávaní odtlačkov vytvára úsečky medzi markantami. Veľkosti úsečiek a uhly medzi nimi sa podieľajú na výslednom porovnaní skóre. V poškodenom obrázku sú zachované všetky pôvodné markanty, do ktorých poškodenie nezasahovalo, na rozdiel od rekonštruovaného obrázku,

kde zmeny markantov nastávajú aj v iných oblastiach. VeriFinger potom vyhodnocuje aj najmenšie zmeny uhlov markantov a znižuje výsledné porovnávacie skóre. Obrázok 8.3 ukaže detegované markanty a úsečky medzi nimi. VeriFinger vyhodnotil tieto dva odtlačky ako zhodné s hodnotením 769 bodov. Ďalší obrázok 8.4 je výsledok porovnania rovnakého odtlačku s rekonštruovaným odtlačkom, ktoré dosiahlo hodnotenie 732. Veľká chýbajúca časť odtlačku je zrekonštruovaná avšak mnoho markantov bolo pozmenených. Veľký počet zmien markantov zníži celkové hodnotenie porovnania odtlačkov a nová zrekonštruovaná oblasť nevyváži svojou hodnotou tieto veľké rozdiely.



Obrázok 8.3: Porovnanie originálneho a poškodeného odtlačku programom VeriFinger.



Obrázok 8.4: Porovnanie originálneho a rekonštruovaného odtlačku programom VeriFinger.

## Kapitola 9

# Záver

Cieľom tejto práce bol návrh a implementácia metódy pre rekonštrukciu poškodených častí odtlačkov prstov. Problematika odtlačkov prstov bola naštudovaná a popísaná v druhej kapitole. Nasledovali metódy zvyšovania kvality odtlačkov prstov, ktoré zahŕňali normalizáciu, kontextové filtre, niektoré ochoreniami pokožky, výpočet kvality odtlačku a možnosti verifikácie odtlačkov. Popis fungovania neurónových sietí je súčasťou nasledujúcej kapitoly. V ďalšej kapitole boli vysvetlené metódy generovania syntetických odtlačkov prstov pomocou matematických modelov ale aj pomocou neurónovej siete. Po návrhu a implementácii metódy rekonštrukcie odtlačkov prstov bola vykonaná analýza získaných výsledkov a možné rozšírenia práce. Všetky body zadania diplomovej práce boli teda splnené.

Bolo vytvorených 5 sád poškodených odtlačkov prstov, na ktorých boli vykonávané testy. Štyri sady sú syntetické odtlačky vytvorené pomocou programu Anguli, piata sada obsahuje reálne odtlačky získané optickým snímačom. Do odtlačkov bolo vkladané poškodenie vo forme syntetickej bradavice o rôznych počtoch. Vytvorená neurónová sieť bola naučená rekonštruovať tieto poškodené odtlačky. Bolo vytvorené jednoduché grafické užívateľské rozhranie, pomocou ktorého je možné rekonštruovať vytvorené odtlačky.

Analýza spočívala v meraní kvality odtlačkov pred a po rekonštrukcii. Výsledky analýzy ukázali že rekonštrukcia odtlačkov pomocou neurónovej siete je možná. Najväčšie navýšenie kvality po rekonštrukcii, meranej pomocou nástroja FiQiVi, predstavovalo +43,5 % pre sadu odtlačkov s desiatimi bradavicami. Zvýšenie kvality odtlačku bolo pozorované na všetkých sádach. Vo väčšine prípadov nastalo zlepšenie kvality aj podľa NFIQ2, tieto výsledky boli ale pomerne nekonzistentné. V prípade sady s jednou bradavicou bola nameraná väčšia kvalita poškodeného odtlačku ako odtlačku bez poškodenia. Ďalším kritériom úspešnej rekonštrukcie je porovnávacie hodnotenie rekonštruovaných a poškodených odtlačkov voči pôvodnému odtlačku. Najlepšie pozorované zlepšenie porovnávacieho skóre dosahovalo hodnoty 6,5 %, pre sadu syntetických odtlačkov s desiatimi bradavicami. Meranie porovnávacieho skóre prebiehalo pomocou nástroja VeriFinger. Testy ukázali že je možné zvyšovať toto skóre avšak na úkor kvality rekonštruovaného odtlačku. Testovanie na reálnych odtlačkoch dosahovalo horšie výsledky aj kvôli nekonzistencii a malému počtu vzoriek.

Táto práca ponúka má množstvo možných rozšírení, medzi ktoré patria napríklad použitie inej architektúry neurónovej siete, optimalizácia pre reálne odtlačky, rekonštrukcia iných ochorení alebo poškodené odtlačky. Za zmienku stojí aj preskúmanie možností neurónových sietí v detekcii poškodenia odtlačku, ktoré je čiastočne možné pomocou navrhnutej neurónovej siete a jednoduchého filtra.

# Literatúra

- [1] ANIL, J., PATRICK, F. a ARUN, R. *Handbook of Biometrics*. 1. vyd. Springer, 2008. ISBN 978-0-387-71040-2.
- [2] CHOLLET, F. *Building Autoencoders in Keras* [online]. 2016 [cit. 2019-12-5]. Dostupné z: <https://blog.keras.io/building-autoencoders-in-keras.html>.
- [3] CHOLLET, F. *Deep Learning with Python*. Manning, 2017. ISBN 9781617294433.
- [4] COLE, S. *History of Fingerprint Pattern Recognition*. 1. vyd. Springer, 2004. ISBN 978-0-387-95593-3.
- [5] DRAHANSKY, M., DOLEZEL, M., URBANEK, J. a ET.AL. Influence of Skin Diseases on Fingerprint Recognition. *Journal of biomedicine & biotechnology*. Máj 2012, roč. 2012. DOI 10.1155/2012/626148.
- [6] DRAHANSKÝ, M. *Fingerprint Recognition Technology - Related Topics*. Lambert Academic Publishing: Lambert Academic Publishing, 2011. ISBN 978-3-8443-3007-6.
- [7] DRAHANSKÝ, M., DOLEŽEL, M. a ORSÁG, F. *Biometrie*. Computer Press a.s., 2011. ISBN 978-80-254-8979-6.
- [8] GERON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1. vyd. O'Reilly Media, 2017. ISBN 978-1-491-96229-9.
- [9] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. *Deep Learning* [online]. MIT Press, 2016 [cit. 2020-1-10]. <http://www.deeplearningbook.org>.
- [10] GOODFELLOW, I., POUGET ABADIE, J., MIRZA, M. a et al. Generative Adversarial Networks. *ArXiv*. 2014, DOI arXiv:1406.2661.
- [11] GURNEY, K. *An Introduction to Neural Networks*. Bristol, PA, USA: Taylor & Francis, Inc., 1997. ISBN 1857286731.
- [12] HALE, J. *Deep Learning Framework Power Scores 2018* [online]. [cit. 2020-2-24]. Dostupné z: <https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a>.
- [13] HARITSA, J., ANSARI, A. H., WADHWANI, K. a et al. *Anguli: Synthetic Fingerprint Generator* [online]. [cit. 2019-12-10]. Dostupné z: <https://dsl.cds.iisc.ac.in/projects/Anguli/>.

- [14] HONG, L., WAN, Y. a JAIN, A. *Fingerprint image enhancement: algorithm and performance evaluation*. 1998, roč. 20, č. 8, s. 777–789. ISSN 1939-3539.
- [15] INOVATRICS. *IDKit Fingerprint SDK [online]*. [cit. 2020-2-13]. Dostupné z: [http://files.innovatrics.com/datasheets/Innovatrics\\_Datasheet\\_Fingerprint\\_SDKs.pdf](http://files.innovatrics.com/datasheets/Innovatrics_Datasheet_Fingerprint_SDKs.pdf).
- [16] ISO. *Biometric sample quality*. ISO 29794-1:2016. Geneva, Switzerland: International Organization for Standardization, 2016.
- [17] KANICH, O. *Výzkum v oblasti simulací poškození otisku prstu [online]*. 2019. Dizertační práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav inteligentních systémů. Vedúci práce Martin Dražanský. [cit. 2020-06-02]. Dostupné z: <http://hdl.handle.net/11012/187324>.
- [18] KANICH, O. a DRAŽANSKÝ, M. *Synthetic Fingerprint Damage Simulator and Generator - Touch Damage Module - Software*. 2014. Dostupné z: <https://www.fit.vut.cz/research/product/600/.cs>.
- [19] KANICH, O., ORAVEC, T. a DEJMAL, D. *Fingerprint Quality Visualizer - Software*. Dostupné z: <https://www.fit.vut.cz/research/product/607/.cs>.
- [20] KISHAN, M., CHILUKURI, M. a SANJAY, R. *Elements of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1997. ISBN 0-262-13328-8.
- [21] LE, Q. *A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks [online]*. 2015 [cit. 2020-1-15]. Dostupné z: <http://robotics.stanford.edu/~quocle/tutorial2.pdf>.
- [22] LEE, D. *Common pitfalls in machine learning [online]*. 2018 [cit. 2020-01-14]. Dostupné z: <http://danielnee.com/2015/01/common-pitfalls-in-machine-learning/>.
- [23] MALTONI, D., JAIN, A., MAIO, D. a AL. et. *Handbook of Fingerprint Recognition*. 2. vyd. Springer, 2009. ISBN 978-1-8488-2254-2.
- [24] MORIN, M. B. a WILLETTS, M. *Non-Determinism in TensorFlow ResNets*. *ArXiv*. 2020, DOI arXiv:2001.11396.
- [25] NEUROTECHNOLOGY. *VeriFinger [online]*. [cit. 2020-02-19]. Dostupné z: <https://www.neurotechnology.com/verifinger.html>.
- [26] NIELSEN, M. *Neural Networks and Deep Learning [online]*. Determination Press, 2015 [cit. 2019-12-5]. <http://neuralnetworksanddeeplearning.com/>.
- [27] O'SHEA, K. a NASH, R. An Introduction to Convolutional Neural Networks. *ArXiv e-prints*. November 2015. DOI arXiv:1511.08458.
- [28] RIAZI, S., CHAVOSHIAN, S. a KOUSHANFAR, F. *SynFi: Automatic Synthetic Fingerprint Generation*. Február 2020. DOI arXiv:2002.08900.
- [29] SERGEY, I. a CHRISTIAN, S. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. In: 2015. DOI arXiv:1502.03167.
- [30] SHERVIN, M. a AMIRALI, A. *Finger-GAN: Generating Realistic Fingerprint Images Using Connectivity Imposed GAN*. December 2018. DOI arXiv:1812.10482.

- [31] TORRES, A. *Origin and History of convolution* [online]. 2010 [cit. 2019-12-9]. Dostupné z: <https://www.slideshare.net/Alexdfar/origin-adn-history-of-convolution>.
- [32] TROZAK, D., TENNENHOUSE, D. a RUSSELL, J. *Dermatology Skills for Primary Care*. Humana Press, 2006. 59-65 s. ISBN 978-1-59259-906-6.
- [33] VARILE, M. *Train neural networks using AMD GPU and Keras* [online]. 2019 [cit. 2020-1-5]. Dostupné z: <https://towardsdatascience.com/train-neural-networks-using-amd-gpus-and-keras-37189c453878>.

## Príloha A

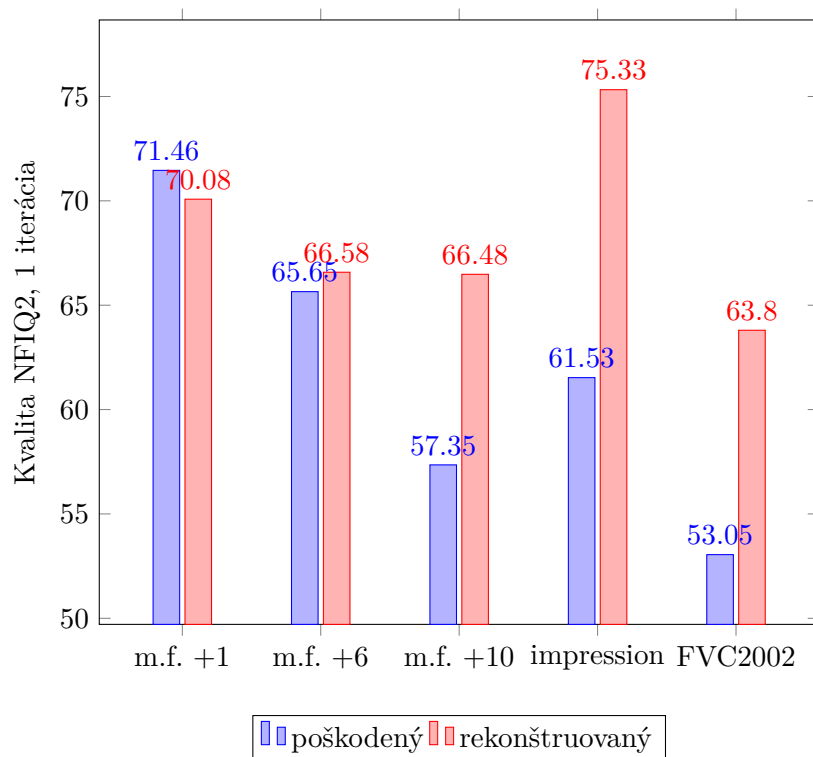
# Obsah priloženého pamäťového média

Priložené pamäťové médium obsahuje:

- skript pre vytvorenie poškodených odtlačkov prstov,
- skript na učenie siete,
- grafické užívateľské rozhranie,
- testy výsledkov rekonštrukcie,
- vytvorené sady odtlačkov,
- súbor README.TXT s podrobnejším popisom obsahu.

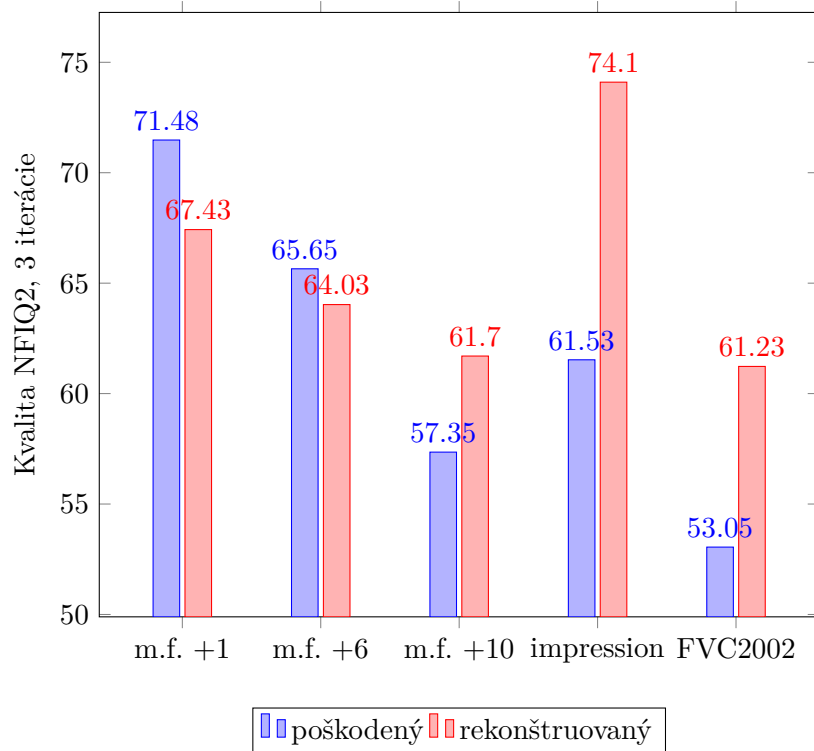
## Príloha B

# Grafy úspešnosti rekonštrukcie

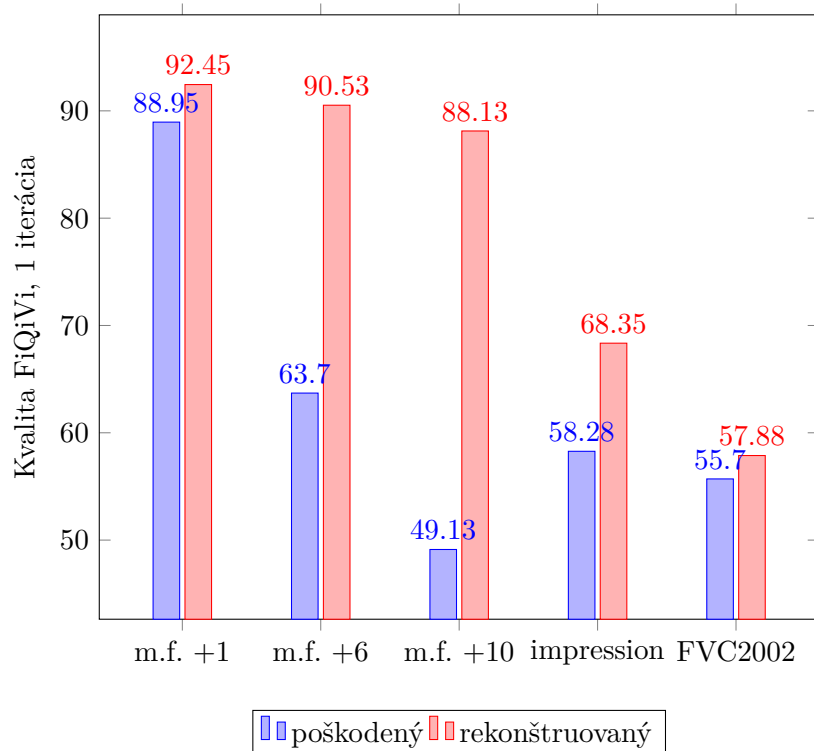


Graf B.1: Priemerná kvalita odtlačkov, meraná pomocou nástroja NFIQ2, 1 iterácia. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.

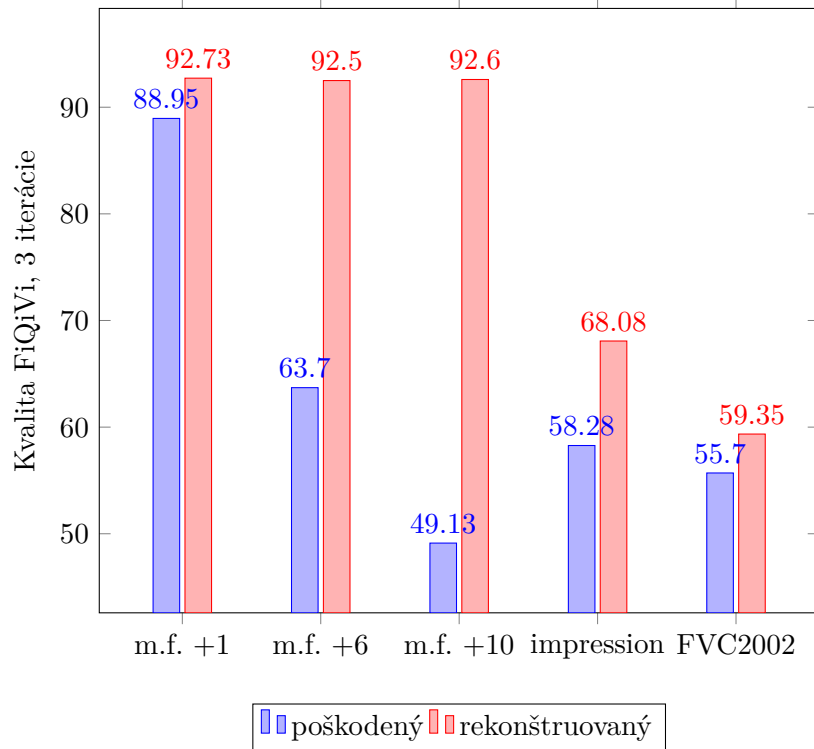




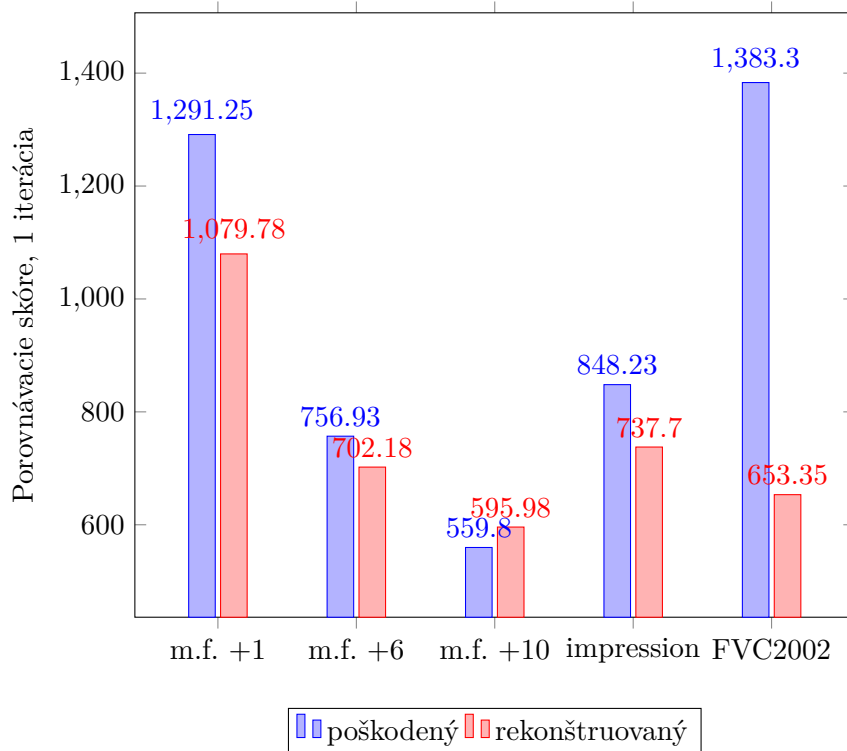
Graf B.2: Priemerná kvalita odtlačkov, meraná pomocou nástroja NFIQ2, 3 iterácie. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.



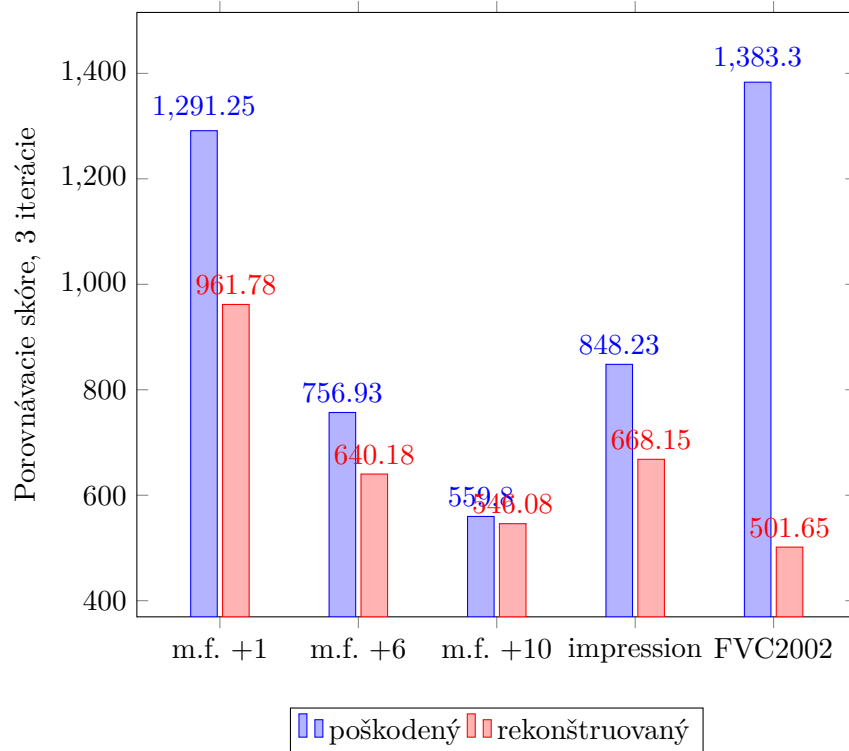
Graf B.3: Priemerná kvalita odtlačkov, meraná pomocou nástroja FiQVi, 1 iterácia. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.



Graf B.4: Priemerná kvalita odtlačkov, meraná pomocou nástroja FiQiVi, 3 iterácie. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.



Graf B.5: Priemerná porovnávacie skóre odtlačkov, merané pomocou VeriFinger, 1 iterácia. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.



Graf B.6: Priemerná porovnávacie skóre odlačkov, merané pomocou VeriFinger, 3 iterácie. Sady: *master fingerprint* (m.f.) + 1, 6 a 10 bradavíc, *impressions* a FVC2002.